Behavior Monitoring Using Visual Data and Immersive Environments

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Joshua Swenson Fasching

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Nikolaos Papanikolopoulos

August, 2017

Acknowledgements

My advisor Nikos, has been a constant source of support and advice through my time in graduate school. His guidance and insight have allowed me to discover capabilities I was previously unaware of.

I would especially like to acknowledge my fellow colleague, Nicholas Walczak. He has always been someone I can go to for technical discussion and advice.

Thoughout my research I have also had the opportunity to work with several talented people directly including: Mackenzie Mikkelsen, Alex Asenbrenner, Rafael Seferyan, Elizabeth Harris, and Austin Young. My fellow graduate students in the lab have also been a constant source of support and advice. Brett Hemes, Dimitris Zermas, Panagiotis Stanitsas, Xinyan Li, Ruben D'sa, William Toczyski.

Abstract

Mental health disorders are the leading cause of disability in the United States and Canada, accounting for 25 percent of all years of life lost to disability and premature mortality (Disability Adjusted Life Years or DALYs) [1]. Furthermore, in the United States alone, spending for mental disorder related care amounted to approximately $201 billion in 2013 [2]. Given these costs, significant effort has been spent on researching ways to mitigate the detrimental effects of mental illness. Commonly, observational studies are employed in research on mental disorders. However, observers must watch activities, either live or recorded, and then code the behavior. This process is often long and requires significant effort. Automating these kinds of labor intensive processes can allow these studies to be performed more effectively.

This thesis presents efforts to use computer vision and modern interactive technologies to aid in the study of mental disorders. Motor stereotypies are a class of behavior known to co-occur in some patients diagnosed with autism spectrum disorders. Results are presented for activity classification in these behaviors. Behaviors in the context of environment, setup and task were also explored in relation to obsessive compulsive disorder (OCD). Cleaning compulsions are a known symptom of some persons with OCD. Techniques were created to automate coding of handwashing behavior as part of an OCD study to understand the difference between subjects of different diagnosis. Instrumenting the experiment and coding the videos was a limiting factor in this study.

Varied and repeatable environments can be enabled through the use of virtual reality. An end-to-end platform was created to investigate this approach. This system allows the creation of immersive environments that are capable of eliciting symptoms. By controlling the stimulus presented and observing the reaction in a simulated system, new ways of assessment are developed. Evaluation was performed to measure the ability to monitor subject behavior and a protocol was established for the system's future use.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Early intervention is a key aspect of treating developmental disorders: the sooner a medical professional can intervene, the higher chance of a positive outcome. However, by the time symptoms are severe enough to be brought to medical attention, the most ideal window for intervention may have passed. Hence, identifying precursors to developmental disorders has received great interest by the mental health field; risk markers that indicate an elevated risk for development of a disorder. Risk markers form a broad category that can include genetic, social, and/or behavioral deviations from the norm.

Determining and detecting risk markers is a long and arduous process, requiring careful observation of subjects, both control and developmentally impaired. These observations are either made live in real-time by a trained individual or recorded for later annotation. With real-time observations, quality is impaired by the need to make annotations concurrent with the behaviors being exhibited. Recordings help to alleviate this problem, but require time consuming manual annotations which can be impractical. Both methods also suffer from the problem of inter-rater variance: different individuals may score the same markers differently.

Many advances in recent years have been made in the use of computer vision for human activity monitoring. Methods have been introduced for face detection [3] and tracking [4] with robust and fast performance. With the leveraging of GPUs deep-learning of convolutional neural networks [5] has lead to great improvements in image classification and detection allowing for robust detection of people and their surrounding workspaces [6]. Tracking of humans in video and other sensors is a well studied topic

in computer vision both in images [7] as well as using multi-modal sensors [8]. While computer vision methods such as these have been validated on larger data sets, they still need to be tested and improved in different and challenging application domains. A challenging application domain for computer vision is in mental health assessment. Simply observing a subject using computer vision may not be enough if risk markers do not present themselves during the observation period.

1.1 Motivation

Computer vision research has started to examine situations in which different techniques can be applied to risk marker detection [9, 10, 11, 12]. Often times visually observable risk markers are rare in occurrence. This can lead to both problems in data set collection as well as missed opportunities for detection if the symptom doesn't manifest itself during the observed period. The primary method in which symptoms of mental illness are assessed are through questionnaires answered by either the clinician, patient or persons familiar with the patient's ailment [13, 14]. This depends on several factors including: the nature of the illness, ability to describe particular symptoms and age of the patient. Visually observable risk markers and symptoms may not manifest themselves over an observation period. However external stimulus from the environment and situations in which the subject is participating may induce symptoms to occur. For instance, in the case of excessive handwashing compulsions in relation to obsessive compulsive disorder (OCD), the environment and situation of washing hands at a bathroom sink can trigger compulsive behavior. This presents an ecological approach, recreating situations and surroundings, to mental health assessment with the desire to capture potential risk markers and symptoms in a timely manner.

Virtual reality and augmented reality systems offer a medium for enabling this ecological approach by providing reconfigurable ways in which to engage patients effectively. An effective system must be able to present a symptom eliciting environment while allowing the subject to interact with the environment in a natural way. In recent years commercial depth sensors and head mounted displays (HMD) have become prevalent to the point of enjoying commercial success outside the research community. HMDs offer an immersing virtual reality experience while not providing much in the way of user

input. Commercial depth sensors provide a means of real-time 3D sensing allowing for natural user interaction. However, in order to use both of these technologies together in a natural way these technologies need to be registered to each other. Once registered together this opens up the possibilities for greater immersion in augmented/virtual reality (AR/VR) experiences displayed in the HMD by making use of a RGB+D sensor with known solutions for full body tracking [15].

This technology promises great advances in the area of human-machine interaction with a plethora of emerging applications. A significant advantage of this approach is its potential to combine the best of two technologies providing a result that offers simultaneously visual stimulation and physical interaction inside the virtual world. Without the use of additional hardware such as wearable sensors or external interest point markers, the participant will be able to experience interactions more naturally. Efficient techniques in human activity monitoring using vision will be required order to enable this interaction as well as further analysis.

1.2 Thesis Statement

An application of this combined RGB+D sensor and HMD system is in mental health assessment. A subject wearing an HMD displaying an augmented reality environment can be elicited particular stimuli through the HMD. Their viewpoint with regards to the stimuli can be measured using the HMD and their movements tracked by the RGB+D sensor. A rudimentary version of this has been performed without the immersive qualities of the proposed system [16, 17]. This thesis seeks to present a way in which symptom triggering stimulus can be provided in a controlled setting which can then be observed by a computer vision system. In effect this closes the loop for assessment by providing a way to track responses to precisely controlled stimuli. This allows for a new framework for mental health assessment where the clinician can work with their patients through constructed virtual experiences and collect precise data on how they react to those experiences. Certain conditions will be more amenable to this framework than others. Traditionally a clinician would either have to recreate the situation in the real world or have the patient recall their past experiences for assessment. This thesis also presents new ways of interacting in virtual environments using a computer vision system that

does not rely on worn devices or an articulated model for part of the subject's body.

This work, in an effort to improve on the current state of the art, aims to prove the following thesis: *Immersive environments and interaction aided by computer vision are a valid avenue for exploration in mental health assessment.*

1.3 Approach

In order to explore the merits of using immersive environments and computer vision for mental health assessment the following tasks were performed:

- Reviewed the relevant literature with regards to mental health assessment, computer vision, activity recognition, and immersive environments in Chapter 2.
- Explored the applicability computer vision to mental health assessment by using techniques to distinguish different symptoms known to co-occur with autism in Chapter 3.
- Investigated how environment and place can be used in mental health assessment as well as developing techniques to make observations about the activity being observed in Chapter 4.
- Designed and developed a system for allowing arbitrary virtual environments to be used and interacted with for mental health application domain in Chapters 5 and 6.
- Validated the measurement capabilities of an instance of the system with one user over several trials and established a protocol for the system's future use in Chapter 7.

1.4 Contributions

The contributions that will be presented in this thesis are described here:

- Established an assessment of the state of the art in computer vision, behavior imaging and AR/VR for mental health treatment and assessment.

- Demonstrated the applicability of computer vision methodologies to the mental health domain by being able to classify symptoms related to autism in video.
- Developed a method for assessing OCD related behaviors which used the scenario and environment around the subject to elicit those behaviors.
- Developed a method for registering RGB+D sensors to HMD systems with minimal additional modification.
- Presented a method for extending densely sampled trajectory features for use in scene flow data.
- Provided a framework for further development of mental health assessment scenarios using immersive environments with natural interaction.

Chapter 2

Literature Review

In this section background material for several different aspects of this work is discussed ranging from virtual reality, to mental health assessment, to the computer vision and machine learning techniques that can be used to create an understanding of observed symptoms. The categorization serves to delineate between the diverse topics investigated.

2.1 Visually Observable Mental Health Risk-Markers

Risk markers and symptoms for mental illnesses manifest themselves in different ways some of which are observable visually allowing for non-intrusive screening. Observing these can be one source of diagnostic information as part of an involved assessment. This is especially true in cases where risk-markers can be described in an objective and quantifiable way. Visually observable risk markers and symptoms are present for several mental illnesses including: obsessive compulsive disorder (OCD), schizophrenia and autism spectrum disorder.

Obsessive-compulsive disorder is an impairing anxiety disorder that affects 2-3% of the adult population. The disease is characterized by having unwanted, intrusive thoughts (obsessions) and/or ritualistic behaviors (compulsions) whose purpose might be to counteract the obsessions [14]. Like most mental illnesses, it is currently assessed

through different clinical measures that take the form of interviews, checklists and self-reporting. In children, one of the most well known measures is the Children's Yale-Brown Obsessive Compulsive Scale (CY-BOCS) [18] checklist. Different aspects of this disorder are visually observable. Some patients have a compulsion to ordering, arranging or symmetrizing items in a particular way [19]. For others, obsessive and meticulous hand washing is a compulsion. These are behaviors that are sometimes also exhibited by a normative population and observing the differences between populations may serve to delineate a difference.

Psychiatric researchers have reported being able to distinguish in video between a preschizophrenic subject and those who would not [20, 21]. Walker *et al.* [20] assessed home movies of subjects (infancy to age 15) who would later be diagnosed as well as healthy siblings and family members. The videos were coded by raters, blind to the diagnosis, looking for examples of: neuromotor abnormalities, motor skills, and infant motor skills. This study found a significant correlation between high frequency of observations of neuromotor abnormalities and the eventual diagnosis of schizophrenia. Neuromotor abnormalities found in the preschizophrenic subjects included: spastic movements, postural abnormality of the trunk or legs, and musculoskeletal abnormalities. In [21], a longitudinal study used videos of Danish children (ages 11-13) recorded in 1972 along with information about the psychological outcomes of the subjects 20 years later. The researchers found that the subjects who were eventually diagnosed with schizophrenia had scored lower on their socialability rating. The difficulty with such an assessment however, is in the subjectivity of social behavior. Observation of neuromotor abnormalities are perhaps the best candidate for automatic observation.

There are neuromotor precursors to autism spectrum disorder as well. In psychiatric literature, the term *motor stereotypies* typically describes a class of actions performed by both normally developing and at-risk children which are: purposeless, repetitive, and suppressible. While these behaviors occur in both normal and at-risk populations they are still of interest to the psychiatric community, as better understanding is essential for early diagnosis and treatment [22]. Goldman *et al.* [23], specifically investigate motor stereotypies exhibited by children with developmental disorders. In their study they found that children with autism had a greater number and variety of stereotypies when compared against children with other development disorders. Singer *et al.* [24]

attempts to provide a more precise description of the kinds of stereotypies. A common theme in these works is an acknowledgment that motor stereotypies still have a very elastic definition but that their presence as an observed behavior warrants further investigation in the context of understanding their link with neurological disorders. The visually observable risk-markers and symptoms discussed here only represent a portion of what can be observed and viable as a tool in psychiatric assessment. In all of the work discussed in this subsection, the visual observations were recorded manually. The following subsection discusses automatic approaches via computer vision.

2.2 Understanding Human Activity in Video

Being able to automatically identify visually observable risk-markers requires methods to understand human activity. Several methods have been suggested for identifying salient interest points in video. One popular approach, [25], finds space-time interest points using a 3D version of the Harris corner detector at different scales along a video volume. Willems *et al.* [26] uses an approach which involves computing the Hessian at each point along the space-time cube and using the determinant of the Hessian to denote saliency, or interest points in the video for further processing. Other approaches take an even simpler route. For instance, in [27] Castrodad employs a temporal differencing scheme across all of the images in a video, with interesting points then detected by retaining space time volumes that exceed a certain threshold. In each method a decision is made to select only the most salient of regions. Dense sampling is an alternative to reduce the number of feature points in a data-agnostic way, where interest points are sampled at regular intervals at a resolution less than the original spatio-temporal resolution of a video [28, 29].

Given space-time interest points, different methods have been employed to describe the appearance and motion present at or around the interest points. Histogram of Oriented Gradients (HOG) [30] and Histogram of Oriented Flow (HOF) [31] features are used alone and in combination by Laptev in [32]. HOG effectively captures appearance information in a compact way by storing gradient orientation information on an image in histograms. Optical flow methods estimate the apparent motion present between images which are encoded as a histogram to make HOF.

In [27], Castrodad uses only the raw intensities of the temporal difference values, yielding decent results, likely due to the hierarchical dictionary learning framework employed. Willems *et al.* [26] compute a modified version of SURF features on their detected interest points. An exhaustive summary of methods is not given but a recent survey on human activity recognition can be found in [33].

Wang, H. *et al.* [29] examine some of these interest-point detector and feature description combinations in a human activity classification context. The standard bag-of-words model was used to describe the video sequences. The bag-of-words model compactly describes a large quantity of feature descriptors by assigning each to different "bags" of representative descriptors and relating this via a histogram. Bag-of-words and alternative pooling, encoding and normalization approaches are explored by Wang, X. *et al.* in [34].

Recent success using deep convolutional neural networks for image classification [5], scene-labeling [35] and object detection [6] has inspired their investigation for use in activity description in video. A natural extension is to apply this convolution to not only space but time as well to create 3D convolutional networks as was done in [36, 37, 38]. A challenge with this approach is finding enough labeled data to properly train such a network. In [39] the relationship of encoding temporal information for activity classification in the context of convolutional networks was explored. They found that while improvements could be gained through properly incorporating temporal information, the traditional CNN approach, when matched to activity classes, still performed competitively. Optical flow-based features are often used to augment these different approaches such as in [37, 40, 38, 41] to improve performance. Their use varies between network architectures. CNNs have been examined in the more complex task of detecting an activity both spatially and temporally as well. In [41] a two stream approach is used to detect activities. Being able to identify and localize human actions in video is still very much an active research topic.

2.3 Behavior Imaging

In [42], Laptev *et al.* used space-time interest points as correspondences when identifying and segmenting out periodic motion from a scene. Wang, P. *et al.* [43] use HOF features on interest points to examine the quasi-periodic nature of actions in social games performed between parents and their infant children. Other works have observed periodic motion in video using a more holistic approach. Cutler *et al.* [44] present an early work on examining self-similarity occurrence maps for the purposes of periodic motion analysis. Junejo *et al.* [45] expands on this by developing a view-invariant descriptor for the purposes of action recognition.

In [10], Rehg *et al.* monitor a professional administering the Rapid ABC exam for Autism Spectrum Disorder to a child subject. The work examines methods for predicting the subject's engagement in the activity using audio and visual information. While the Multi-Modal Dyadic Database (MMDB) used in that work is publicly available, it does not contain examples of the stereotypies examined in the initial findings of this thesis. Hashemi *et al.* [9] focus on engagement of the subject being examined in a protocol similar to the Rapid ABC. Their method focuses on modeling the appearance of body part landmarks on the subject's face to measure the exact orientation of the face relative to an object. Their work also examines a separate risk-marker of gait asymmetry in a subject using body pose tracking.

Other works have also examined stereotypies using computer vision. In [12], Ciptadi *et al.* tackle the task of recovering a queried stereotypic activity performed by the same actor in other sequences. Their work focuses on the activities: jumping up from chair, jumping on the floor, and paddling movement of the hands, whereas our work focuses on activity classes with more subtle motion patterns (see Figure 3.2). Rajagopalan *et al.* [11] use the detector and descriptor of [25] to classify motor stereotypies in a data set comprised of YouTube videos of children at various ages. We compare our methods on the data set of [11] as well as our own data set.

2.4 AR/VR and Mental Health

Many VR systems using HMDs need to employ some form of calibration in order to be used in a natural way. Perhaps the most relevant work related to this thesis are

systems which need to localize an HMD relative to some camera. An early system by Kato *et al.* [46] used fiducial markers to calibrate an HMD for a VR video conferencing system. These markers also served as a reference point for calibrating their camera to the system. State of the art methods use infrared optical tracking as well as on board sensing from the HMD to achieve very accurate results, with many commercial systems already available [47]. These systems however, can be quite expensive and are often not portable. Furthermore while these systems can be used for localizing both an HMD and an RGB+D sensor, they require extra equipment and calibration of their own systems. Therefore focus should be on systems which directly incorporate the RGB+D sensor.

There are a few examples that bring RGB+D or Depth sensors and HMDs together. However these systems remove the challenge of localizing the HMD to the depth sensor by rigidly attaching them together. Suma *et al.* [48] use a motion capture system to track the HMD inside of a large room size workspace. The points from the rigidly attached RGB+D sensor are then projected into the virtual world using the known configuration of the HMD as sensed by the motion capture system. The Ovrvision product [49] is a stereo vision system capable of being mounted on an HMD. It offers both see-through HMD capabilities and the possibility for leveraging stereo vision for interaction.

The tracking methodology for the HMD sensor used in our experiments is discussed in [50]. That work does not include the position camera and LEDs that were introduced in the Rift DK2 version. It is believed that this tracking system uses actively modulated LEDs on the HMD to provide unique signatures for the position camera to identify points on the HMD. Doing this simplifies the registration procedure allowing for fast acquisition of the HMD by the camera.

Consumer-grade yet powerful solutions for natural user interaction such as the Microsoft Kinect and ASUS Xtion along with middle ware such as the Microsoft SDK and NITE have set a new bar for the field. These solutions have an advantage in that they don't require the user to wear any additional components to accomplish interaction. The more sophisticated parts of the middle ware, such as human pose tracking [15], do make assumptions on the user based on their learned models. There have been attempts to move away from these assumptions and still provide meaningful user interaction.

In [51] calibrated RGB+D input is used to crudely estimate scene flow (pixel-wise velocity in $\mathbb{R}^3$) using image-based optical flow [52] combined with depth information.

This provides an estimate of point-wise forces acting on objects. They also present a scheme for interpreting grasps of 3D objects in the presence of these forces. This is used in conjunction with their see-through AR system called HoloDesk. The entire system does not take into account recent advances in HMD technology and their sensing methodology makes the assumption that all interaction will occur in the fixed workspace of the HoloDesk. Scene flow-based force interaction for AR was extended in [53] to a CAVE environment using multiple calibrated RGB+D sensors to provide model-free interaction for an entire human body.

Having an immersive and portable VR system can further enhance our understanding of mental health. One area where VR has found relatively large interest is in exposure therapy. In fact it has had over ten years of study [54]. Exposure therapy involves patients confronting anxiety triggering stimuli in a controlled environment as a way to lessen the fear effect overtime. VR based exposure therapy has been shown to be effective in treating PTSD caused by combat related stress [55, 56, 54]. Potential treatments for anxiety and phobia related disorders using such techniques have also been explored with promising potential [57]. VR has been used in management of persistent pain as well. Schroeder *et al.* [58] use VR along with haptics to create different user experiences to shift the subject's focus away from pain. They explored three different virtual environments for managing pain. Their approach also provides feedback to the user that guides them through the session. This work takes advantage mobile phone-based VR solutions that are more portable and cost effective. While these examples of VR's use in mental health have focused on treatment, assessment has been explored as well to a lesser degree.

What follows are systems that have used VR to further understand mental illness by simulating environments in a controlled way to elicit a response. Some studies have simply used a computer monitor and a standard PC interface to elicit responses offering the lowest level of immersion.

Van den Heuvel *et al.* [59] subjected participants to 'clean' and 'dirty' visual stimuli on a monitor and measured their responses using a PET scan. This was used to localize structures in the brain that could be influencing the OCD-diagnosed participant's symptoms. In a pilot study Simon *et al.* [60] attempted to create a standard set of images and videos that provoke a strong response for a wider range of obsessions and

compulsions.

More recent approaches have begun to fully incorporate virtual reality. Kim *et al.* [17] used an HMD, with rotational tracking, and a joystick to simulate and interact a virtual environment as part of an obsessive-compulsive disorder (OCD) study. The participants were asked to complete two sets of tasks in the virtual environment as prompted by a virtual display. They were then asked to go back and check on their accomplishment of the first set of tasks. Using data from the HMD as well as their position in the VR world, the study tracked the frequency of checking behaviors. They claim that this correlated positively to self-reporting of the subjects (OCD and healthy controls). A summary of their entire work and a survey on VR for OCD research can be found in [16].

Nolan *et al.* [61] have also taken what they refer to as a "ecological approach to neurophysical testing" by using virtual reality to simulate everyday environments. In their study, they replicate the VIGIL-CPT [62] exam for testing attention and inhibition in subjects. The subjects are situated in a virtual classroom in which the stimuli that they are to focus on is displayed on a virtual whiteboard. Various distractions of the classroom are also simulated. Through the pose of the HMD they are able to gauge how well the subject is performing the test. They found that the method had similar effectiveness to the VIGIL-CPT exam. However due to the limited sample size they could not confidently make an assessment about how attention improves with the age of the subject. Several other studies [63, 64, 65] have made use of the virtual classroom paradigm of further evaluating its used for studying attention and inhibition in subjects.

Some works have started to incorporate physical interaction into their virtual environments for mental health assessment. Parsons *et al.* [66] propose a system in which the user is presented with a virtual grocery store that they must navigate and collect items to assess various mental abilities. To enhance immersion they propose tracking the hand of the user using a infrared based tracking system. The system registers that the hand has grabbed something by detecting a gesture. The physics of the hand object interaction however are not modeled.

Our proposed system would enhance works of the style described above by incorporating marker-less depth sensing as a natural input medium.

2.5 Literature Summary and Limitations

Visually observable symptoms for different mental illnesses do exist and computer vision researchers have started to apply research from their discipline to these challenges. A majority of the work so far has been in parsing behaviors associated with Autism Spectrum Disorders (ASD). The use of virtual reality environments for mental health assessment has been explored to some degree as well. However in these cases, input hasn't come from natural movement but from mouse or joystick. VR and AR systems are increasingly becoming better at presenting users with configurable immersive environments. This requires precise calibration with sensing and display components of their systems. Furthermore natural methods of interfacing using these systems have yet to be applied to mental health assessment.

The state of these current methods provides the basis for this thesis. While each of these components may work well in their own domain their connection to each other warrants study.

Chapter 3

Activity Classification

This chapter describes the approaches explored for classifying different motor stereotypies as they appear in video sequences. It serves as an exploration into using computer vision techniques in mental health assessment. Providing classification methods for known neuromotor abnormalities is an important step in a detection pipeline. A view-invariant feature is described which is amenable for use in conjunction with multi-view tracking systems. As a comparison, a state-of-the-art feature description intended for single views is also presented. Two methods for classifying the presented feature descriptions for video are discussed and briefly compared.

3.1 Feature Description

This section covers the different feature descriptions explored during the initial findings. The first is a viewpoint invariant feature description amenable to inclusion in multi-camera systems such as [67]. The other is a state of the art single viewpoint feature description.

3.1.1 Log-Polar Histogram Features

Several feature descriptions have been suggested for encapsulating the essential information of video data for action classification. This subsection provides a description of a view-invariant feature description proposed by Junejo *et al.* [45].

Designing a view-invariant feature description requires being able to describe similar information from multiple camera viewpoints. This particular description is based on the observation that, while actions may appear different due to viewpoint changes, the recurrence of the appearance remains similar. A frame-wise self similarity matrix (SSM) can capture this recurrence and is the basis for the feature description described in this section.

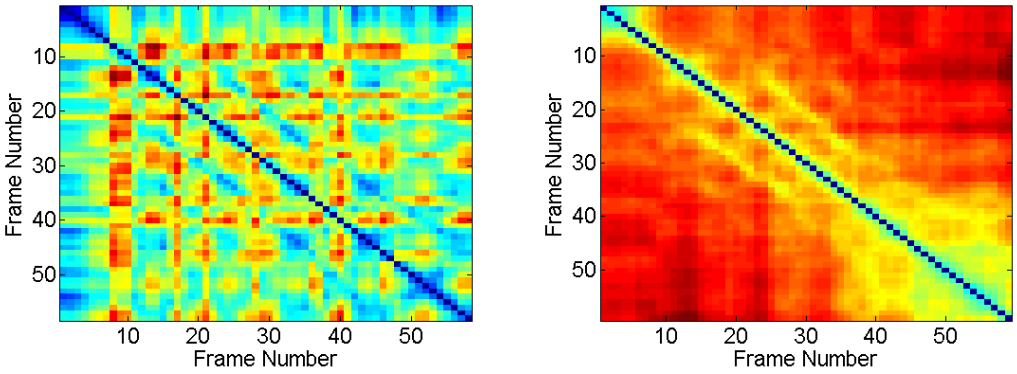
For a given n_v frame long video sequence $V = \{v_i\}_{i=1}^{n_v}$, where $v_i \in \mathbb{R}^{W \times H}$ is a width $W \times$ height H image frame, a self-similarity matrix S is computed by first computing a frame-wise feature description on each v_i. The self-similarity matrix is then constructed on the feature descriptions by performing an all-by-all frame comparison using the Euclidean distance between vectorized versions of the feature descriptions (referred to as d_i). This can be represented as

$$S(i, j) = \|d_i - d_j\|_2, \tag{3.1}$$

where $S(i, j)$ represents the element in the i^{th} row and j^{th} column of the $n_v \times n_v$ self similarity matrix for V.

Two different frame-wise feature descriptions were chosen for investigation. The first, optical flow, was chosen since stereotypic behavior will likely create motions on the image plane which will repeat with time. Optical flow was computed using the method of Bruhn *et al.* [68]. Figure 3.1(a) depicts an optical flow self-similarity matrix for the hand-flapping action. Here, one can see the quick and repetitive change of the optical flow between frames associated with this particular action. In addition to optical flow, we also incorporated the popular HOG [30] descriptor, to add an appearance-level feature description. Figure 3.1(b) shows an example of the self-similarity matrix of HOG features between frames.

In order to compactly describe a self-similarity matrix the log-polar histogram (LPH) approach was used. A semi-circular window of radius r is situated at a particular frame i on the diagonal of a self-similarity matrix. The window is aligned along the diagonal. It is split up into three equally spaced concentric regions, with the last two sections being partitioned into five equally angularly spaced regions. This creates $n_w = 11$ regions inside of the window. Inside of each region a histogram of $n_h = 8$ bins discretizing the directions of the gradient of the self-similarity matrix is computed. Areas of the



(a) SSM of Optical Flow (b) SSM of HOG

Figure 3.1: Example self-similarity matrices for the hand-flapping stereotypy. The degree of similarity between frames is color coded with dark blue for most similar varying to dark red for least similar.

window that fall outside of the matrix are assigned a gradient of zero. The histograms are normalized such that the bins sum to one. Concatenating the histograms together leads to an $n_h \cdot n_w$ dimensional feature vector encapsulating the self-similarity matrix for a particular window size for a particular frame. Since the window in which an action is performed is not known ahead of time, the LPH descriptor is computed for different window sizes and then concatenated to form the final descriptor. The window sizes used were $r = \{7, 14, 28\}$.

Each frame-wise feature description yields its own self-similarity matrix over the course of an action. They can easily be fused together by concatenating their resulting LPH descriptors. However for computational efficiency reasons, it is also important to examine their performances separately. In [45], different features are incorporated together using a multi-channel kernel for classification, however concatenation was chosen here to allow for comparison between the chosen classification methods.

Performing the aforementioned process for each frame and on each frame-wise feature self-similarity matrix yields a set of LPH descriptors for a video sequence. This can then be used as input to a classification framework.

3.1.2 Densely Sampled Trajectory Features

This subsection serves as a brief overview of the densely sampled trajectory (DST) features [28] explored for classifying motor stereotypies in video. Unlike the previous LPH features, DST features have largely been used for data sets containing actions performed from a single view.

The tracked points from which the trajectories are derived are initialized at regular pixel intervals $J = 5$ on the image frame. This sampling occurs at several different scales to establish the densely tracked points. Dense optical flow is computed between frames in the video sequence. The next location of the tracked image points is then dictated by the flow vectors of the optical flow. Post-processing is done to remove potentially erroneous tracked points. Static trajectories and trajectories containing extremely large variations in position between frames are pruned. Additionally, the length of the trajectories is limited to a fixed amount of frames $T = 15$. This ensures that each tracked point is indeed the same feature point across time, otherwise the errors can accumulate over time and possibly lose track of the point.

The remaining trajectories are themselves used as a feature. For a trajectory of frame length T, there is a sequence of displacements on the image plane I, $D = (\Delta I_t, \Delta I_{t+1}, ..., \Delta I_{t+T-1})$. This is then normalized by the l_2-norm to yield D'. This process in effect encodes the shape of the trajectory. This feature will be referred to as the dense trajectory feature.

These trajectories can also guide sampling for additional features. Each trajectory can be used to create a spatio-temporal volume by sliding an $N \times N$ window centered on the trajectory across time. This volume can then be further separated into subvolumes by discretizing the image space into $n_\sigma \times n_\sigma$ blocks and the temporal space into n_t blocks. This creates $n_t \times n_\sigma \times n_\sigma$ subvolumes along each trajectory in which local features are computed. In the experiments the parameters $N = 32, n_\sigma = 2, n_t = 3$ were used.

As with the LPH descriptor, the local features are HOG and optical flow. Each subvolume consists of T/n_t images of size $N/n_\sigma \times N/n_\sigma$. Low-level image features are computed on each of these images as normal (image gradients for HOG and optical flow for HOF), however instead of binning these features across solely the image plane, they are also binned across time. This means that the HOG or HOF feature for a particular space-time subvolume will incorporate image gradient or optical flow information from

several frames.

The final descriptor is computed by concatenating the histograms from each space-time subvolume to create the dense HOF or dense HOG feature (depending upon the image feature used). The dense HOG and dense HOF histograms are l_2-normalized as is done in [30] and [31].

Optical flow is also used to compute motion boundary based features [31]. In this case, the components of optical flow between a pair of frames is split into feature images U and V. The gradient orientation is computed on both images independently to identify the boundary of motion changes. The intended effect is to negate consistent motion that would not be removed with simply using only the dense optical flow. As with HOG and HOF features, motion boundary histograms MBH_x and MBH_y corresponding to feature images U and V are computed in each space-time volume along a trajectory and normalized in the same way. MBH_x and MBH_y are concatenated by space-time subvolumes and then by feature to form the dense MBH feature.

This process leads to a dense set of features for a given video. Therefore a rich set of information is given as input to a learning framework.

3.2 Classification

Effective feature descriptions should capture discriminating aspects of the data they are describing. These representations then need to be interpreted. What follows are two classification methods used to interpret the feature descriptions in an efficient way. Depending on the classification method additional preprocessing methods may need to be applied.

3.2.1 Dictionary-Based Classification

A dictionary is a set of basis vectors that can be linearly combined into a weighted sum to describe another vector. Recent work, such as [69] and [70], has gone into efficiently learning dictionaries that describe a data set of vectors well, using a weighted sum of a small subset of basis vectors.

In [71] Guha and Ward explore different methods for classification using dictionaries determined for sparse representation of a signal. The first method explored is very

similar to the bag-of-words classification approach. A dictionary is learned over the entire training set that explains the training set using a sparse weighted sum of the vectors in the dictionary. The activated coefficients for each example feature in a video are then counted and stored as a histogram summarizing these activations over a video for that feature set. These histogram features are then used as the input for training an support vector machine (SVM) classifier. As the authors of [71] point out, bag-of-words can be related to this method by restricting the sparse recovery to a single vector, also known as vector quantization. However, the most successful approach examined focused on learning dictionaries that sought to represent each class. This is the method that is examined here.

A set of n_k feature descriptors $X_k = \{x_{j,k}\}_{j=1}^{n_k}$ can be computed from video sequences of subjects exhibiting an action class k. A dictionary D_k is determined for each class k that solves the problem

$$\underset{\alpha_k, D_k}{\text{minimize}} \quad \sum_{j}^{n_k} \|x_{j,k} - D_k \alpha_{j,k}\|_2^2 + \lambda \|\alpha_{j,k}\|_1. \tag{3.2}$$

Classification of a query video, with n_Q features $X_Q = \{x_{j,Q}\}_{j=1}^{n_Q}$, is determined by selecting the class k that solves

$$\underset{k, \alpha}{\text{minimize}} \quad \sum_{j}^{n_Q} \|x_{j,Q} - D_k \alpha_{j,k}\|_2^2 + \lambda \|\alpha_{j,k}\|_1, \tag{3.3}$$

for some fixed sparsity penalty λ.

Given an arbitrary data set it is difficult to choose the appropriate λ as well as dictionary size. Therefore several possible values are tried and the parameters are chosen that best meet the objective. The same parameters are used for each class.

Equations (3.2) and (3.3) are solved using the online dictionary learning framework of Mairal *et al.* [70]. Solving Equation (3.3) is often referred to as the LASSO problem and is solved using the Least-Angle Regression (LARS) algorithm of [72] and is known to be very efficient. For details on how Equation (3.2) is solved, see [70].

3.2.2 Support Vector Machine (SVM)

In addition to classification by dictionary-learning, SVM classification using the one-vs-one approach for multi-class classification was considered. The one-vs-one approach

was used for multi-class classification with K classes where $K(K-1)/2$ classifiers are constructed to separate pairs of classes. The one-vs-one SVM classification problem can be formulated as

$$\begin{aligned} \underset{w^{ij},b^{ij},\zeta^{ij}}{\text{minimize}} \quad & \frac{1}{2}(w^{ij})^T(w^{ij}) + C\sum_t(\zeta^{ij})_t \\ \text{subject to} \quad & (w^{ij})^T\phi(x_t) + b^{ij} \geq 1 - \zeta_t^{ij}, \text{ if } x_t \text{ is in class } i, \\ & (w^{ij})^T\phi(x_t) + b^{ij} \leq -1 + \zeta_t^{ij}, \text{ if } x_t \text{ is in class } j, \\ & \zeta_t^{ij} \geq 0. \end{aligned}$$

Here, x_t are the training examples, w is the linear separator, b is the offset and ζ_t are the slack variables. $C > 0$ is a tunable parameter which expresses the trade-off between the slack variables and the margin objective. The implementation provided by [73] for solving this problem was used.

Rather than use the features described in Section 3.1 directly, the popular bag-of-words framework [28, 29, 32, 45] is used to create histograms of features that are used for classification. Features from a video are vector quantized, sum-pooled and normalized by l_1 to create a histogram. The dictionary used for vector quantization comes from computing K-means on a subset of the training data. The number of clusters is determined experimentally for each feature. It is important to note that using the bag-of-words approach removes the temporal association between features in much the same way that learning class dictionaries does.

Due to the histogram features used, the χ^2-kernel , defined as

$$k(x,y) = \exp\left(-\gamma\frac{1}{2}\sum_i\frac{(x_i-y_i)^2}{x_i-y_i}\right), \tag{3.4}$$

was used to transform the data supplied to the SVM algorithm. This is a popular kernel for bag-of-words feature data and was used in several works related to action recognition including [28], [29], and [32]. The kernel is closely related to the χ^2 distance measure between frequencies.

Tuning parameters C, the SVM tradeoff parameter, and γ, the kernel scaling parameter, are selected by performing a grid search on the two parameters and selecting the pair that maximizes the accuracy on 5-fold cross-validation on the training set. Cross-validation is performed by partitioning the training set into n equal folds (sections) and

Total Number of Subjects	29
Number of Samples Per Action	
Hand Washing	37
Ear Covering	78
Hand Flapping	38
Shoulder Shrugging	115
Head Shaking	71
Total	339

Table 3.1: Data set summary.

holding out each fold for testing, while the classifier is trained on the remaining folds. In other works, γ is set heuristically to the inverse of the mean distance between training examples. However we found that this approach was not competitive with searching for γ for either motor stereotypy data set.

3.3 Results

The methods discussed in the previous sections were applied to two video data sets. These data sets capture different action classes as well as different scenarios for recording while being in the same problem domain.

3.3.1 Laboratory School Data set

Experiments were performed on a data set that consists of videos recorded in a pre-school classroom in order to test the performance of the various methods described previously. Videos were recorded with IRB approval at the Shirley G. Moore Laboratory School; A summary of the data set can be found in Table 3.1. Since the occurrence of the behaviors intended for classification can be rare even in a clinical setting, a "Simon Says"-like mimicry game was employed to elicit behaviors in normal children that approximate motor stereotypies. The game consists of a leader who performs a particular action and child participants who are instructed to mimic the actions performed by the leader.

The leader was instructed to perform actions that approximate the following stereotypic behaviors: ear covering, hand flapping, hand wringing/washing, shoulder shrugging and head shaking (see Figure 3.2). However, pre-school children are not always

good at doing as they are told and so adherence is poor. Only suitable examples of the desired actions were included in the data set. Some actions such as "ear covering" are shorter and were repeated several times throughout the course of the game, which is one source for differing numbers of samples across classes. In addition to children not performing the actions, the variance in how each child performs each action is large. This is due in part to developing motor skills but also because the children are easily distracted by the antics of their peers.

In order to perform classification, bounding boxes for each participant were manually annotated, providing generous bounds to account for the high variability in actions. In effect, this assumes ideal person detection, since the primary focus of the work is on action classification. This is similar to other activity classification data sets with a single actor in each clip.

Given the size of the data set, the methods were validated using Leave One Person Out cross validation. For each individual subject, a set of classifiers was learned using examples that did not come from that particular subject performing the action.

Each feature description was examined with both classification methods as well as with select feature combinations that were guided by tractability and time constraints. Figure 3.3 summarizes the accuracy for each combination tested. The results presented in that figure use the optimal parameters for the appropriate method which were found experimentally. For the dictionary-learning based classification approach the number of dictionary atoms and the sparsity penalty parameter were varied. SVM-based classification required selecting the number of words for the bagging procedure. Figure 3.4 shows the effect of changing the number of words. It can be seen that the optimal word size is clearly different for the LPH features and the DST features.

Figures 3.3, 3.5 and 3.6 show that the best performing method is learning the Dense Trajectory features using an SVM with a χ^2-kernel. Figure 3.5(b) shows the discriminative ability of the classifier across all the action classes. This is advantageous as the feature descriptor size for the Dense Trajectory can be smaller than that of the other features depending on the size of n_σ and n_t.

Generally speaking, ear covering was the most difficult to classify amongst all options, being most often confounded with shoulder shrugging. This could be due to the shoulders of the subject move as part of ear covering. Other pairs of classes such as

(washing/flapping) and (shrugging/head shaking) were confounded to a lesser degree. This is possibly due to similar motion.

While the view-invariant LPH features perform worse, they are still able to classify a few action categories as evidenced in Figure 3.6. These results are improved by combining the view-invariant features for appearance and motion into a single feature, at least when using the dictionary learning based framework as seen in Figures 3.6(a) and 3.6(b). Figure 3.6(c) shows that the recovery of the shoulder-shrugging action improves with the combination of features.

3.3.2 Self-Stimulatory Behaviors Database

The SSBD is a collection of 75 videos from the YouTube.com website depicting three different behaviors (Arm Flapping, Head Banging, and Spinning). Two of these behaviors (Arm Flapping and Head Banging) were explored in the previously discussed data set as (Hand Flapping and Head Shaking). The labels used for performance analysis on this data set are kept consistent with [11] for comparison. Unlike the previous data set, these videos vary widely in duration, kind of camera being used, setting, and camera pose. The videos are not clipped strictly to the activity being displayed. While the data set provides these annotations, they were ignored to keep with the protocol presented in [11]. Leave One Group Out cross validation was performed where 5 randomly selected videos from each group were left out from training for each fold in a 5-fold procedure (3 for 10-fold). The procedure for choosing parameters C and γ were identical to those used for the first experiment. The trackers for the DST Features were initialized at $J = 25$ pixels to allow for computations to be tractable.

Table 3.2 provides a comparison of the methods explored in this work with respect to [11]. The results for [11] in Table 3.2 were performed using their code and data. They were performed independently to avoid ambiguities in the cross-fold validation assessment. Note an improvement in performance over [11] when using the Dense Trajectory features. This suggests that such behaviors might be best characterized in video by the point-wise trajectories of the appendages in question for the stereotypic behavior. Somewhat surprisingly, the LPH features perform slightly worse than the others examined. Since the videos depict the subjects at different viewpoints the features should exhibit additional robustness. However since these videos are not necessarily clipped

some examples can have lighting or viewpoint changes which can confound the repetition of features required for the SSM. While both the methods presented here and the results in [11] have a high standard deviation, it is important to keep in mind the high degree of variability in the data set itself; the videos come from Youtube and cover broad examples of the action categories.

n-fold (%, σ)	[11]	SVM+DT	SVM+HOG	SVM+HOF	SVM+MBH	SVM+ALL	SVM+LPH_HOG	SVM+LPH_OF
5	52.0 (10.95)	56 (16.7)	34.7 (5.6)	44 (10.1)	30.7 (18.0)	42.7 (16.1)	48 (8.7)	36 (13.8)
10	28.7 (13.7)	50 (13.6)	36.7 (10.5)	43.3 (22.5)	30.0 (17.2)	45 (24.9)	46.7 (24.6)	36.7 (15.3)

n-fold (%, σ)	[11]	DL+DT	DL+HOG	DL+HOF	DL+MBH	DL+LPH_HOG	DL+LPH_OF
5	52.0 (10.95)	49.3 (12.1)	42.7 (15.4)	54.7 (11.0)	49.3 (11.2)	53.3 (11.5)	38.67 (3.0)
10	28.7 (13.7)	46.7 (21.9)	48.3 (12.3)	50.0 (22.2)	53.3 (10.5)	53.3 (17.2)	40.0 (11.6)

Table 3.2: Classification accuracy results on the SSBD for both classification methods.

3.4 Summary

In this chapter, different methodologies for classifying motor stereotypic behaviors were explored to varying degrees of success. Learning one-vs-one SVM classifiers on the Dense Trajectory feature was found to work best on video data collected from pre-school age children participating in a "Simon Says"-like game where children performed behaviors designed to resemble common stereotypies. The method performs well despite large age-related variability in the actions carried out.

The ultimate goal of this work is to develop this methodology to be appropriate for use in a classroom and/or clinical setting. In the short term, this means incorporating the techniques presented here with an installed system at the Shirley G. Moore Laboratory School. For these environments it becomes necessary to distinguish between what appears to be non-stereotypic behavior and stereotypic behavior as well as being able to classify which behavior is being exhibited. Such a technology can provide an early passive screening mechanism for neurodevelopmental disorders, bringing those potentially afflicted to health care professionals faster to ensure improved prognosis.

Still this is not the only technique from computer vision that can be amenable to mental health assessment. The next chapter explores a situation in which environment, place and task are used to elucidate behaviors in OCD and healthy subjects. Computer

vision methods are presented for automatically assessing part of their performance on these tasks.

(a) (b) (c) (d) (e)

Figure 3.2: Example image sequences for different motor stereotypies. Subfigures depict ear covering (a), hand flapping (b), hand washing (c), shoulder shrugging (d), and head shaking (e). Note the subtlety of actions (d) and (e). Faces blurred to preserve anonymity.

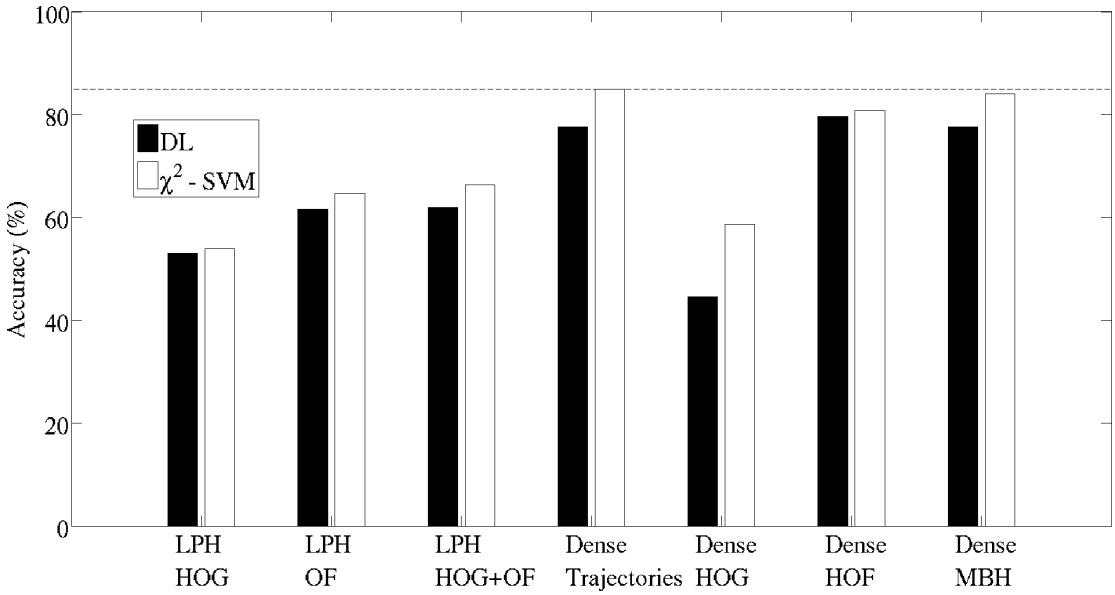


Figure 3.3: A comparison of accuracy rates across different feature combinations and learning methods.

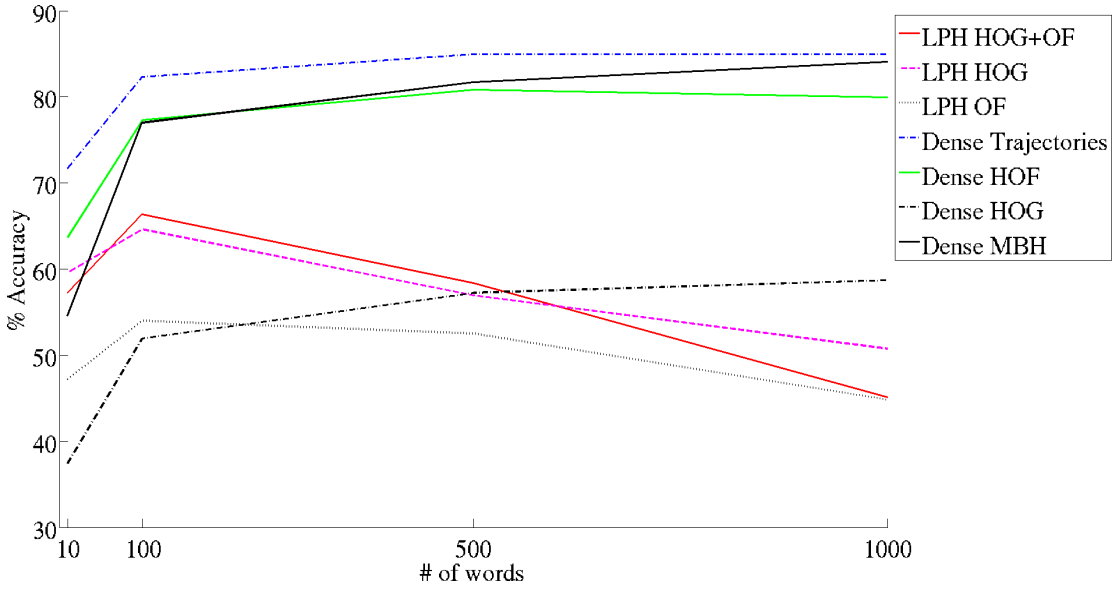
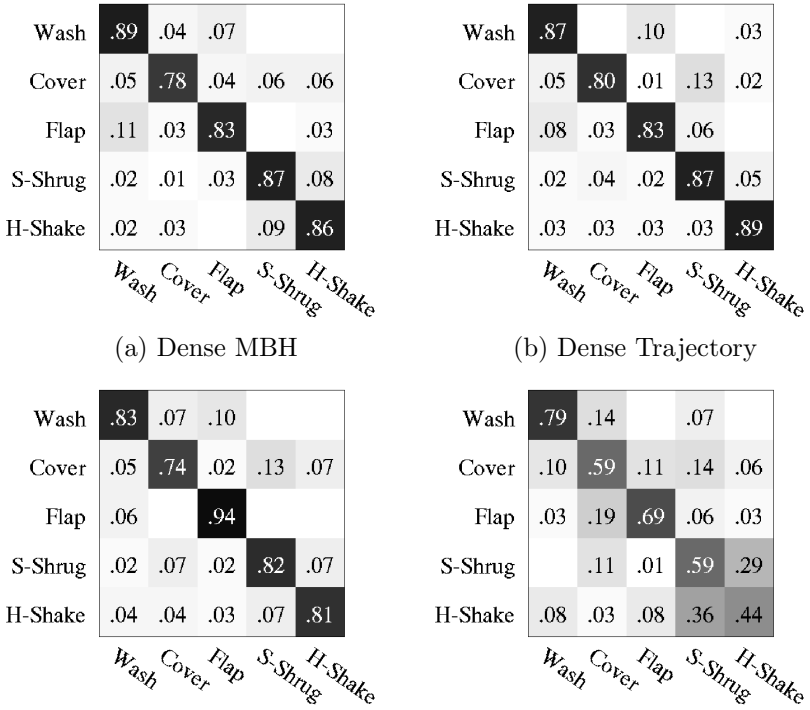


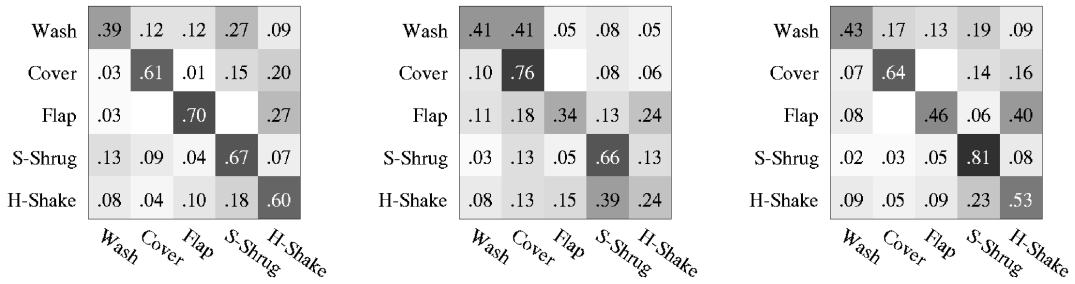
Figure 3.4: Accuracy results for different numbers of words.



(a) Dense MBH (b) Dense Trajectory

(c) Dense HOF (d) Dense HOG

Figure 3.5: Confusion matrices for different DST features classified using a χ^2-kernelized SVM. Each row indicates the distribution for how each class is labeled.



(a) Optical Flow (b) HOG (c) HOG + Optical Flow

Figure 3.6: Confusion matrices for different LPH features classified using the Dictionary-Based Approach. Each row indicates the distribution for how each class is labeled.

Chapter 4

Environments for Mental Health Assessment

This chapter explores an ecological approach to mental health assessment. In the previous chapter certain behaviors were preselected and acted out by participants. While this is useful for showing that such behaviors can be classified, an ecological approach may induce such behaviors to occur naturally. Certain behaviors of obsessive-compulsive disorder (OCD) were examined in verifying this approach.

OCD is an impairing anxiety disorder that affects 1-3% of youth. Children and adolescents with OCD experience debilitating obsessions and compulsions that often disrupt their daily activities and interactions. Many of these behaviors manifest themselves through elements of the physical environment, and yet very little research has focused on understanding how OCD relates to aspects of interior and exterior spaces. Previous research has examined the relations between children with autism spectrum disorder (ASD), which shares some overlapping features with OCD, and their interactions with the physical environment [74, 75]. Much of this research is devoted to implementing design and architectural strategies to enhance the physical environment to better facilitate social interactions and learning in children with ASD. A multidisciplinary study was conducted at the University of Minnesota investigating the environmental factors related to OCD provides the data used for this chapter [76].

The study observes youths with OCD and matched healthy controls engaging in

everyday tasks. The tasks take place in the Travelers Innovation Lab in the College of Design and are videotaped. The automated analysis of such videos can also enable data analysis on a much larger scale. One of the tasks in the study is handwashing, as ritualistic handwashing is a common compulsion observed in individuals with OCD. This work focuses on the analysis of handwashing videos recorded by an overhead camera. Figure 4.4 depicts the handwashing station, which was only one section of the exam room. A method is presented for automatically labeling different steps of a handwashing activity from overhead videos. The labeled steps include: `turnsOnWater`, `turnsOffWater`, `appliesSoap` and `rinsesSoap`. Areas of interest are labeled *a priori* and tracked throughout the activity. Figure 4.1 shows an example frame from one video with the areas of interest annotated. These steps were chosen for their determinability and the availability of annotation from the study. Background subtraction is then used to indicate when these areas of interest become activated. The system is validated by comparing results to hand-labeled ground truth.

This following section provides a detailed explanation of our approach. The results of applying our approach are then presented in Section 4.2 with conclusions and closing remarks following in Section 4.3.

4.1 Approach

4.1.1 Participants

Videos come from a larger OCD study where 18 youths with OCD and 21 healthy controls (males and females), ages 5-17, were studied. Mean age was 11.5 years for OCD participants and 10.7 years for controls. There were no significant differences between groups on age, gender, or ethnicity. Children with OCD had a mean score of 21.8 on the CY-BOCS, which is in the moderate severity range. Of children with OCD, 78% were receiving psychotropic medications and 61% were receiving therapy for OCD.

After obtaining written informed consent and assent, the project coordinator administered the CY-BOCS assessment [18]. Parents completed the Child Obsessive-Compulsive Impact Scale – Revised (COIS-R) and Behavioral Assessment System for Children-2 (BASC-2), and children completed the COIS-R and Multidimensional Anxiety Scale for Children (MASC-2). Participants completed tasks designed by the research

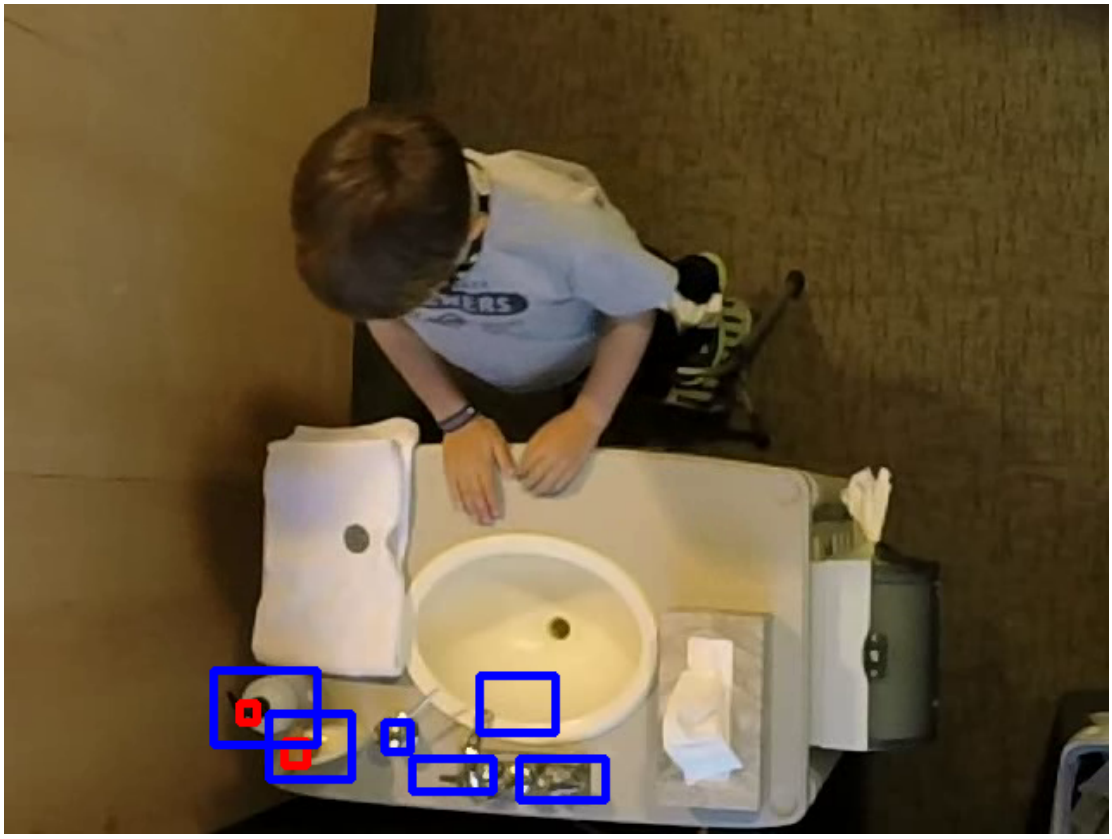


Figure 4.1: Example of an annotated frame from one of the handwashing videos. Note the additional red ROIs for the tracked soap dispenser objects.

team, including preferences of pattern images, free arrangement, arrangement in contrasting environments, and handwashing. The tasks required subjects to interact with their physical surroundings and were expected to elicit observable behavioral differences between OCD and control groups.

This work recorded handwashing at a portable sink and selected data for experiments using automatic annotation. A downward-facing camera was placed above the sink environment to observe the participants. Each video was cropped temporally to include only the handwashing activity and cropped spatially to center the sink. This cropped video was used as input to the automated annotation procedure. The algorithm was used to detect four incident times: when the faucet is turned on (`turnsOnWater`), when the faucet is turned off (`turnsOffWater`), when soap is applied (`appliesSoap`), and

when soap is rinsed (`rinsesSoap`). Each video had regions for soap dispensers, sink, and faucet handles which the authors manually annotated (see Figure 4.1). Not all videos contained all of the annotated regions, but all possible regions are 'soap', 'b_soap', 'o_soap', 'l_handle', 'r_handle', 'sink', 'towel', 'paper_towel', and 'trash'.

4.1.2 Algorithm

The handwashing activity of interest consists of several substeps. Participants are expected to turn on the water, apply soap, lather the soap, rinse the soap, turn off the faucet, and then dry their hands. Comparing these different aspects between healthy controls and subjects with OCD can help determine important differences between the two populations. For instance, a participant with a handwashing compulsion may spend an abnormally long time lathering soap (measured as the time between `appliesSoap` and `rinsesSoap`). Another commonly observed difference occurring for OCD patients is spending extra attention in washing between fingers, both when applying soap and drying. Each substep of the handwashing activity involves interactions with some object in the environment. To apply soap, participants must interact with one of several possible soap dispensers (either a soap dispenser built into the portable sink or stand-alone dispensers on the countertop). Participants must activate the faucet handles to turn on the water, and they must put their hands into the sink bowl to rinse. Therefore, the landmarks can be monitored to see when the participant interacts with them to determine which step of the handwashing procedure the participant is doing.

The observed sink environment is assumed to be stationary and the background is modeled using a multi-layer statistical approach [77]. The background subtraction method uses local binary patterns and a photometrically invariant color measure to build statistical models for each pixel. Given an image sequence, $\{I^t\}_{t=1,\ldots,N}$, the background model at timestep t is defined as $\mathcal{M}^t = \{M^t(x)\}_x$, where x is a pixel in the image. The per-pixel model is defined as

$$M^t(x) \;=\; \{\, K^t(x)\,,\; \{m_k^t(x)\}_{k=1,\ldots,K^t(x)}\,,\; B^t(x)\,\},$$

where $K^t(x)$ is a scalar that denotes the number of $m_k^t(x)$ modes, and the first $B^t(x)$ modes represent stable background observations. Each mode is defined as

$$m_k \;=\; \{\, I_k, \hat{I}_k, \check{I}_k, LBP_k, w_k, \hat{w_k}, L_k\},$$

where I_k is the average RGB vector, $\hat{I}_k$ and $\check{I}_k$ are the estimated maximal and minimal RGB vectors, LBP_k is the average local binary pattern, w_k denotes the weight factor, and $\hat{w}_k$ is the maximal value to which m_k belongs with $k = 1, ..., K^t(x)$. $L_k = 0$ implies the mode does not belong to a stable background layer.

Figure 4.2: Example of a background subtraction result (right) from one frame (left) of a handwashing video. Higher brightness indicates higher confidence that a particular pixel is foreground.

Updating the model creates a background distance map, which is analogous to a foreground probability map. This distance map consists of the distance to the closest mode for each pixel in the input image. If the matching mode does not belong to a known background layer ($L_k = 0$ and $k > B^t(X)$), then the distance is set above the foreground threshold. When two modes are matched, if the distance is above a threshold a new mode is created, otherwise the modes are combined, with a learning parameter affecting how much emphasis is put on the newly matched mode. The distance equation between a pixel and the modes at that pixel location as well as the model update equation can be found in [77].

An example frame from this method is depicted in Figure 4.2. While this background subtraction approach worked well in practice, there are many alternatives; see [78] and [79] for recent surveys.

The foreground probabilities are used to determine when certain regions of the sink area become activated. These regions are annotated *a priori* and represent areas like the soap dispenser, sink faucet, sink bowl, *etc.* Within the annotated ROI, the foreground

score for each pixel is averaged into one value, which yields a noisy indication of when the subject is passing over ("activating") a ROI. This creates an activation signal which can then be analyzed to determine when different substeps of the handwashing activity are performed. Each ROI will generate its own activation signal, generally leading to one activation signal related to each substep. Figure 4.3 shows an example of these activation signals for one particular handwashing video.

Some of the soap dispensers are able to be moved from the sink, breaking the static assumption. These objects are tracked using the TLD tracker [80]. This tracker continuously updates its understanding of the tracked object by maintaining a pair of P-N experts that estimate the number of missed detections and false alarms, respectively. The tracker performs at near real-time rates. These objects have two annotated bounding boxes in the initial frame: one that encompasses the entire object and is updated by the tracker and one that is fixed relative to the object bounding box to encompass the activation point (see Figure 4.1). The activation bounding box is stored as an offset to the object bounding box and maintains a rigid position relative to the object bounding box. In the case of multiple movable objects in the scene, each one is initialized with its own tracker. Interactions between the trackers are not considered. When the object is occluded and tracking is lost—as is likely when it is in use—the last known location and bounding box dimensions are used.

Three different methods were explored for translating the noisy activation signals into measures of substep start and end frames. Each method takes a time series of averaged foreground scores $S_R = \{s_1, s_2, ..., s_T\}$, where T is the number of frames in the series for an annotated region R.

`gm_thresh` method

This method considers sequence entries s_i above a certain threshold to be valid activations. Rather than selecting a threshold arbitrarily, the threshold is learned from the data. A two-mean Gaussian mixture model is learned from one of the sequences. Every entry in S_R is then compared against this model. If an entry in S_R is assigned to the larger mean, then it is considered an activation. Start times are selected when the sequence changes from a non-activation state to an activation. End times are recorded for the converse as well.

`conv` method

This method identifies changes in the sequence S_R using convolution with a 1D Prewitt filter ($kernel\ size = 21$). First, the sequence is smoothed using a moving average filter. The signal is then convolved with the filter to identify changes in the sequence marking an activation. Non-maximal suppression is then applied to single out the largest responses. The remaining responses are stored as start frames for activities. The filter is then flipped and a similar procedure is performed for falling edges and end frames.

`findpeaks` method

This method attempts to locate peaks in each S_R, which then guides selection of the start and end of each peak [81]. The start and end of each peak are converted to the start frame and end frame of an activation, respectively. Peaks in S_R are found by taking the discrete derivative of the sequence and locating zero crossings. Candidate peaks are selected from the zero crossing points if they have a slope in the derivative and amplitude in the original sequence S_R above certain thresholds. A Gaussian function is fitted for each candidate peak around a fixed window ($w = 15$) to establish its shape. The data is assumed to follow the Gaussian function

$$y_w = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x_w - \mu^2)}{2\sigma^2}\right). \tag{4.1}$$

A linear system is formed by taking the logarithm of the previous equation, resulting in

$$A\hat{x} = b \tag{4.2}$$

where

$$\begin{aligned} b &= \left[\log(y_w)\right]^T, \\ \hat{x} &= \left[-\tfrac{1}{2\sigma^2} \quad \tfrac{2\mu}{2\sigma^2} \quad \ln\left(\tfrac{1}{\sigma\sqrt{2\pi}}\right) - \tfrac{1}{2}\left(\tfrac{\mu}{\sigma}\right)^2\right]^T, \\ A &= \left[{x_w}^2 \quad x_w \quad \mathbf{1}\right]. \end{aligned} \tag{4.3}$$

Solving this system yields a σ surrounding a peak, by this conversion,

$$\sigma = \frac{1}{\sqrt{-2\hat{x}_1}}. \tag{4.4}$$

Start and end frames are selected to be at $\pm(1.77)\sigma$, from the candidate peak.

The shape of S_R is interpreted differently by each method. Classifying activation signals in two categories is done by `gm_thresh` , which attempts to distinguish transitions between the two classes. The `conv` method ignores this classification, instead acting directly on transitions detected by the filter. The `findpeaks` method attempts to more rigorously identify the whole transition scenario at once by identifying peaks in the transition and fitting a curve to the portion of the signal around the peak. This is the most computationally intensive method of the set.

In order to compare against the ground truth, the start/end times from the characterized activation signals must be coded into incident times for the four activities of interest: `turnsOnWater`, `turnsOffWater`, `appliesSoap`, and `rinsesSoap`. This was done by creating a set of rules for which start time indicates the beginning of an activity. For instance, the water cannot be turned off before it is turned on and water cannot be turned off before soap is rinsed, so `turnsOffWater` is characterized as the first faucet activation that occurs after `rinsesSoap`. The `appliesSoap` activity is characterized as the first soap activation that occurs. The `rinsesSoap` activity is characterized as the first sink bowl activation that occurs after `turnsOnWater` and `appliesSoap`. The `turnsOnWater` activity is characterized as the first faucet activation that occurs.

4.1.3 Defining Accuracy

As part of the larger study, the videos were coded to indicate the time at which each activity occurs. This coding was performed manually by 3 individuals, and those codes were combined to create a consensus score. These manual annotations were used as a ground truth for comparing against the automated coding methods. Accuracy was calculated by comparing the difference between algorithm-computed and ground truth times for each activity. Algorithm-computed times were deemed correct if they fell within a time window centered on the ground truth time. Accuracy for an activity is defined as the ratio of correctly computed times versus the number of videos containing the activity. The size of the time window was varied in order to characterize the sensitivity of accuracy to the time window. Accuracy was chosen over the more common precision/recall since it is not possible to have a false positive; the number of activities is fixed so there will always be a coded start time for each activity. This can lead to

a false negative if the algorithmically-generated code is not within the time window. However, no false positives can be generated, so precision will remain static as the time window is varied in size.

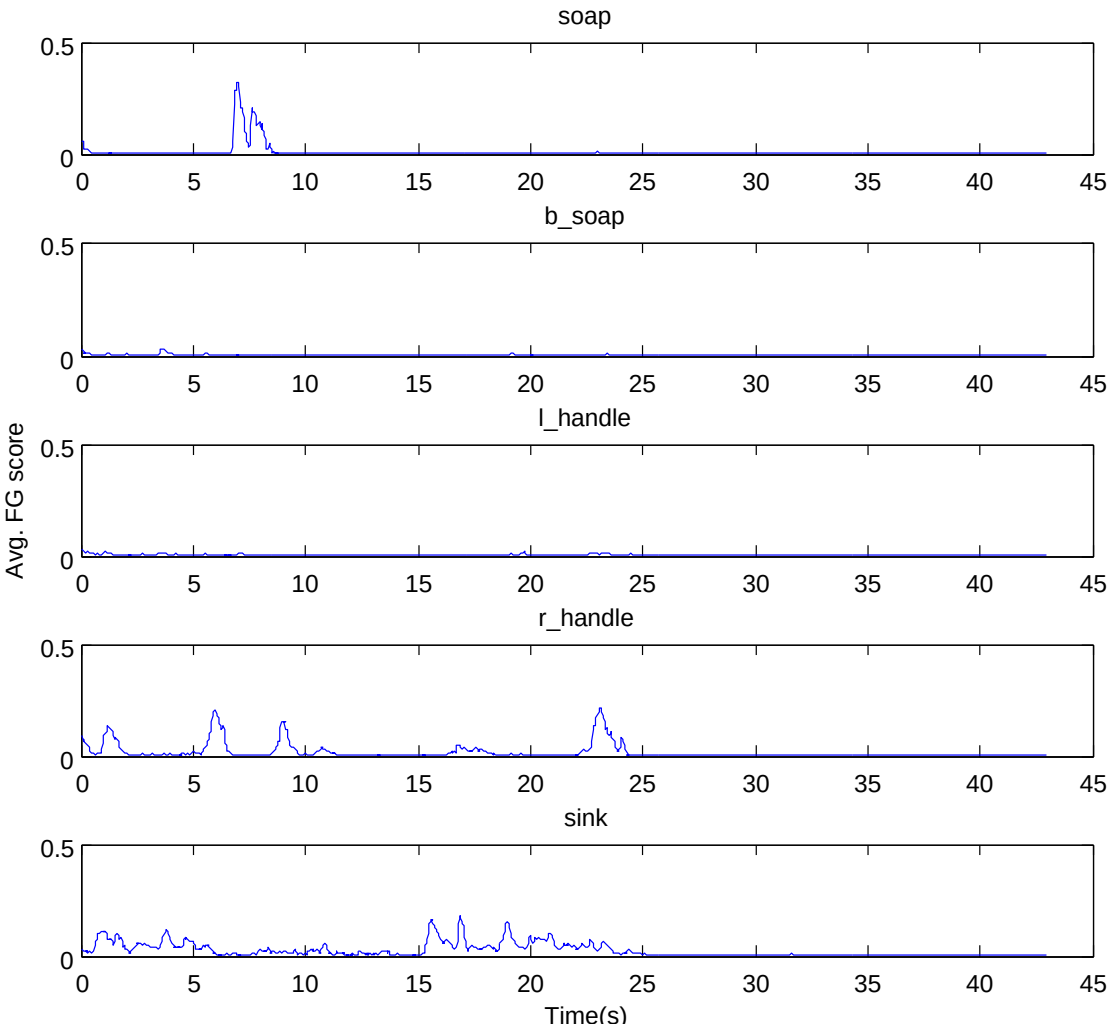


Figure 4.3: Average foreground scores over time for one video. Peaks correspond to the ROI being used. The names 'r_handle' and 'l_handle' correspond to the ROIs for the sink handles from the viewpoint of the rater. The different soap dispensers are referred to as 'soap' and 'b_soap' and are observed as independent ROIs.

4.2 Experimental Results

The methods discussed in Section 4.1 were applied to the 33 videos from the OCD environmental factors study. A single video was recorded for each subject. However, 6 videos were not suitable due to recording errors. Each video was cropped to center the sink at the bottom of the image and provide a resolution of 640x480 at 30 frames per second. Recordings were taken based upon subject availability, so time of day as well as weather conditions vary. Coupled with the sink being placed in front of a large window (see Figure 4.4), this leads to a large variation in illumination both between different videos and often within the same video.

Figure 4.4: Setup of the handwashing station. The stool pictured was non-functioning and placed near the sink to create a more realistic environment. This was deemed important as it may be a trigger for contamination obsessions.

In order to prevent abrupt changes in foreground probability at the beginning of a recording, a background subtraction model was learned for each recording individually,

and that learned model was then applied to the same recording again. Several parameters must also be tuned for the background subtraction. First, the input is scaled down by a factor of 2, since the full resolution is not necessary and the down-sampled images process considerably faster. Additionally, the inputs are smoothed with a Gaussian with a sigma of 0.7. An important parameter is the learning rate for the background subtraction method. It was found that a faster learning rate (0.75) performs far better than a slower learning rate (0.5). Figure 4.5 depicts the performance comparing the faster learning rate (fr) and the slower learning rate (sr).

Accuracy was computed for the activities: `appliesSoap`, `rinsesSoap`, `turnsOnWater`, and `turnsOffWater`. In order to provide a reasonable summary of performance, all accuracy values were averaged across each video in the data set. The time window is varied from 0s to 4s. If a method predicts an occurrence within the time window relative to the ground truth it is considered correct. The accuracy at 0s is always 0%, but this is expected because 0s is below the resolution of the ground truth, so any exact matches within the window would have to occur due to chance. Ground truth was generated by using timestamp information in the video playback rather than frame numbers. Additionally, while disagreement amongst labelers was generally low, it was not uncommon to have several seconds of difference between each labeler. A time window of 2s is the most appropriate as it falls within the margin of error of the ground-truth labelers. A time window of 4s and beyond risks losing too much information.

Figure 4.6 shows a plot of the average accuracy from three different signal characterization methods. To provide a good summary of performance for each activation signal characterization, accuracy is averaged across all activities. From this, it is clear that the `findpeaks` method performs best. Each method also has a sharp drop-off where the accuracy begins to plateau.

Figure 4.7 highlights in more detail the performance of the `findpeaks` method, showing the average accuracy for each activity. From this, detection of `turnsOnWater`, `turnsOffWater`, and `appliesSoap` events all perform well. At a time window of 2s, there is an average accuracy of 90%, 83.3%, and 86.6%, respectively. This approaches the best performance we could hope to accomplish given our assumptions. For instance, in one video the operator enters the scene and shows the subject where different items on the sink top are and touches the different soap dispensers. Given the current assumptions,

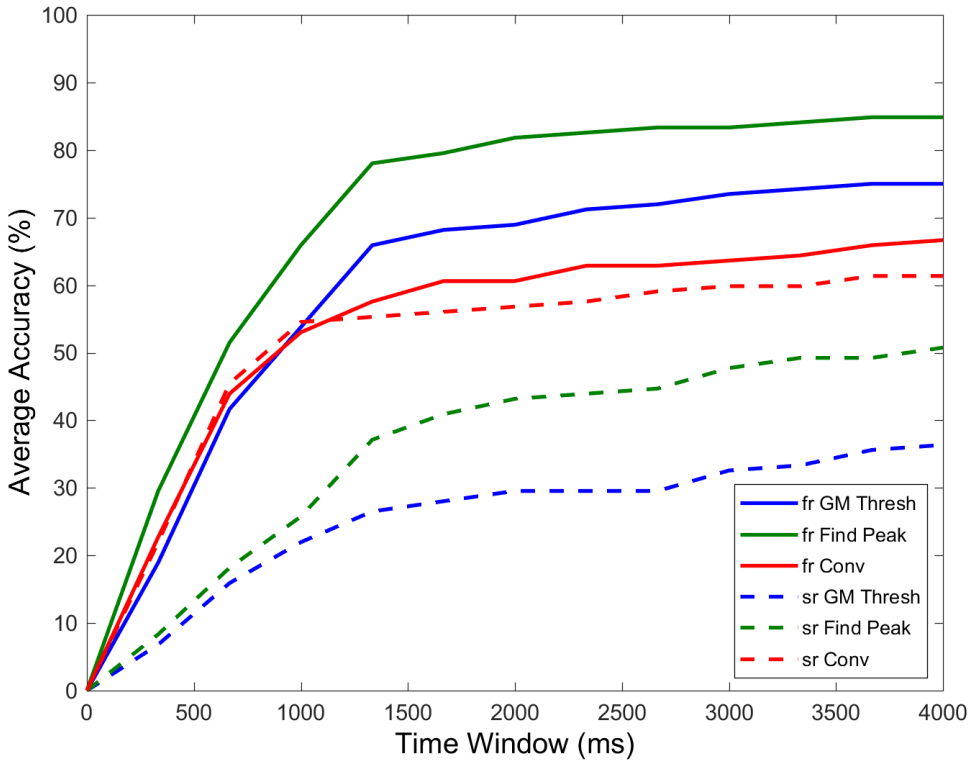


Figure 4.5: Accuracy averaged across each video and across each behavior comparing a slower (sr) and faster (fr) background learning rate.

examples like this will always fail for `appliesSoap`.

While 3 of the 4 activities perform well, the accuracy for `rinsesSoap` barely exceeds 60%. The 'sink' ROI corresponding to the sink bowl is near the 'l_handle' and 'r_handle' ROIs for the sink handles. The subject's hand can trigger the 'sink' ROI when reaching for the sink handles. The participants also cast shadows over the sink that can disrupt the background segmentation. These effects can be seen in Figure 4.3, where a noisy signal for the 'sink' ROI precedes a high activation for 'r_handle'.

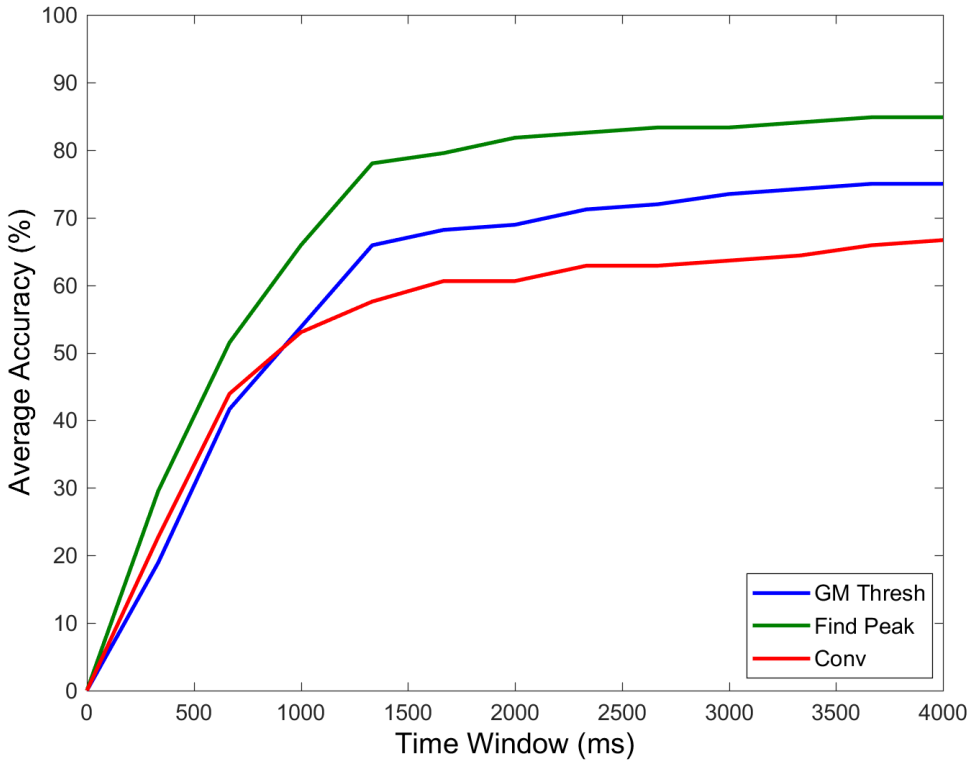


Figure 4.6: Accuracy averaged across each video and across each behavior.

4.3 Summary

This chapter explored the use of computer vision tools as part of an OCD study on how environment and place affect behavior in OCD subjects and healthy controls. In corresponding psychiatric study [76] it was found that longer duration of handwashing was highly correlated to higher scores from the CY-BOCS scores of subjects in the ordering/repeating and forbidden thoughts dimensions. This chapter shows that the time for annotating these handwashing videos can be reduced from hours down to minutes using an automated methodology. Videos of handwashing recorded as part of a study at the University of Minnesota were automatically annotated for start times of different subactivities of handwashing (`turnsOnWater`, `turnsOffWater`, `appliesSoap`, `rinsesSoap`). The automatically generated annotations were compared to hand-labeled

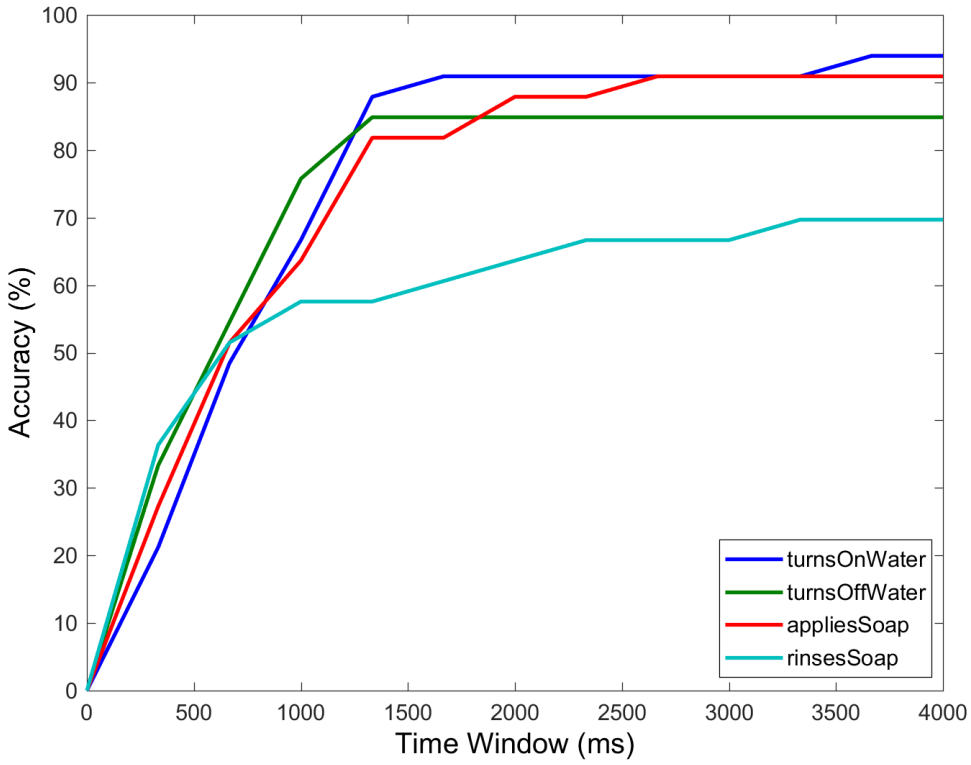


Figure 4.7: Accuracy averaged across each video for the `findpeaks` classifier with each behavior displayed.

annotations to validate the methods. Allowing for a 2 second time difference between ground truth and automatic method, an average accuracy across all activities of 81% was achieved. While there is room for improvement, some failures are difficult to avoid, such as outside actors interfering (a researcher or another individual in the room), or the participant deciding not to fully cooperate because he/she is distracted or for some other unknown reason.

Accuracy on this task can be improved in various ways. One such strategy would be characterizing the different substeps using a hidden Markov model (HMM). Using an HMM can allow a graphical model to be used to add more constraints to the relationship between the handwashing substeps. Another possibility that is popular amongst other works that examine handwashing is to track the hands of subjects within the video.

Additionally, we would like to consider more handwashing activities, particularly

activities related to drying (such as picking up a towel and dropping the towel). There are other aspects as well that were coded manually but do not yet have an automated coding, such as how many towels were used (for disposable towels), how much soap was used (in terms of number of pumps on the dispenser), and other miscellaneous behaviors like if the subject wiped down the sink top with a towel. The data set also includes videos from a frontal view point and the overhead videos that were used. Incorporating these frontal views could provide another avenue for automated coding. These techniques could be applied to the other activities included in the study, especially the free arrange and arrangement in contrasting environments.

Drawbacks of using an ecological approach like the one presented in this chapter are clear. It requires having the setting and equipment in place in order to do the examination. If such setting and equipment are found to be useful in eliciting detectable behavioral makers then it would be desirable to replicate this elsewhere. Doing so for each useful setting and equipment can quickly become expensive. Virtual and augmented reality solutions provide a way in which to deliver these settings in a cost effective way. The following chapters provide detail has to how this can be made a reality.

Chapter 5

Enabling Immersive Environments

The work in this chapter describes the steps taken to enable immersive environments. Section 5.1 describes a methodology for localizing an HMD in an RGB+D sensor's view for the purpose of recovering the RGB+D sensor's relative pose in Section 5.2. This is a crucial first step in being able to use both of these technologies together in a single system. Once this achieved, the next section describes how to use this registered depth data in an interactive environment that is both real-time and natural. The presented system consists of an Asus XTION RGB+D sensor and an Oculus Rift DK2 HMD, however our methodology is designed to be agnostic to the type of HMD and depth sensor used. The HMD has a position camera that uses IR LEDs on the HMD to perform localization and pose estimation. In order to register the two devices, the transformation between the RGB+D sensor and the position camera must be recovered. Since it is assumed the pose of the HMD is provided, if the pose of the HMD can be recovered in the frame of reference of the RGB+D sensor, then the transformation between the RGB+D and position cameras can be computed.

5.1 HMD Localization

The first step is detecting the HMD with the RGB+D sensor, in order to recover the HMD pose in the RGB+D sensor's frame of reference. The HMD detection method

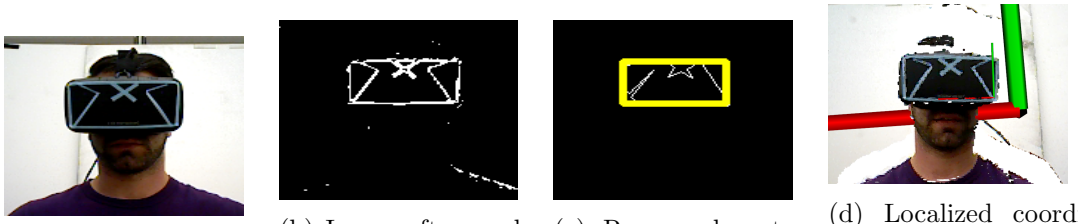
Figure 5.1: Example of the additional markings used on the Oculus Rift DK2 HMD. The placement was chosen to cover the maximum area of the faceplate as well as avoid the active marker LEDs on the device.

must satisfy a number of important constraints, including computational efficiency, consistency, and being independent from the HMD tracking system. Additionally, any solution must meet challenging aspects, such as being robust, smooth, and continuously detecting the HMD.

Our approach makes the assumption of having a controlled environment and consistent illumination conditions. We utilize a color detection scheme in the HSV color space combined with a background subtraction algorithm to localize the HMD in each RGB image. A custom blue mask in a distinct pattern was placed carefully to avoid interference with the infrared LEDs on the front part of the HMD (see Figure 5.1).

A visualization of each step of the HMD detection process is shown in Figure 5.2 and the whole process is summarized in Algorithm 1. Geometric constraints applied on the 3D data from the RGB+D frame are exploited for the fine tuning of the methodology. The software developed utilizes the PCL [82] and OpenCV [83] libraries and the input data consist of RGB and depth video sequences registered to correctly overlap.

Initially, a mixture of Gaussians background subtraction algorithm [78] is applied to remove static portions of the scene from the image (line 6), since the only moving agent in the process is the HMD. Each RGB frame is converted into the HSV colorspace – a hue, saturation, and intensity value is computed for each pixel. Empirically selected



(a) Initial RGB image. (b) Image after applying thresholds in HSV. (c) Recovered rectangle mask. (d) Localized coordinate frame in point cloud.

Figure 5.2: Results of performing each step in the localization pipeline. The two localized frames in (d) are the virtual world frame and the HMD frame.

thresholds in each of the three channels (H,S,V) are used to segment out the distinctive pattern on the HMD mask (line 7) under the constant ambient illumination. The HSV colorspace was chosen for its property to make objects of a particular hue, such as the blue pattern on the mask, stand out. Morphological operations are used to link weakly connected sections in the thresholded image, followed by a blob detection scheme on the resulting binary image, which then detects the largest blob as belonging to the HMD (lines 8-10). An affine bounding box is fit and superimposed upon the blue markings (line 11) resulting in significantly fewer 3D points to be considered in the subsequent steps of the methodology.

The coordinates of the RGB points within the bounding box are used as a mask to extract the position of the front part of the HMD inside the synchronized depth frame. Knowledge of the intrinsic calibration parameters for both the RGB and depth cameras of the RGB+D sensor allows for the accurate registration of the output images and the correct recovery of the 3D points that correspond to the bounding box (line 12).

The geometry of these 3D points forms a bounded planar surface in three dimensions. The centroid of this surface can be used to estimate the translation of the HMD with respect to the RGB+D camera (line 13).

In order to estimate the rotation and validate the other estimates, an artificial planar bounding box of the size of the known HMD faceplate is created. This planar surface is initialized at the centroid of the points encompassing the mask. Instead of being described by a mathematical model, the artificial plane was discretized as a set of 3D points (line 14). The Iterative Closest Point (ICP) algorithm [84] is used to recover a transformation between the artificial plane and the points lying on the fit

Algorithm 1: Pseudocode of the HMD localization process.

Result: ${}^{D}_{S}T$: transformation matrix describing the sensed frame S on the HMD mask w.r.t. the RGB+D sensor frame D.

1 **Initialization:**
2 I : streaming input RGB image from RGB+D sensor
3 D : streaming input depth image from RGB+D sensor
4 **Main Loop:**
5 **while** *Localization == ON* **do**
6 I_{fg} = **RemoveStaticBackground**(I);
7 I_{hsv} = **HSVThreshold**(I_{fg});
8 I_{morph} = **MorphClosing**(I_{hsv});
9 I_{CC} = **ConnectedComponents**(I_{morph});
10 $I_{pattern}$ = **max**(I_{CC});
11 I_{bb} = **BoundingBox**($I_{pattern}$);
12 D_{mask} = **Extract3DPoints**(I_{bb}, D);
13 t = **Centroid**(D_{mask});
14 $ArtificalPlane$ = **GenerateArtificialPlane**(t);
15 ${}^{D}_{S}T$ = **ICP**(D_{mask}, $ArtificialPlane$)
16 **end**

plane model. The result of this procedure gives a measure of ${}^{D}_{S}T$, the transformation matrix describing the sensed frame S on the HMD mask w.r.t. the RGB+D sensor frame D, and a confidence of the fitted transform (line 15). In cases that the ICP algorithm does not converge within a threshold, the initial estimation can be considered invalid and discarded.

This approach satisfies the desired constraints since it provides a consistent detection of the desired object and even validates the results. Furthermore, it is computationally viable, straightforward to implement, and is completely decoupled from the tracking system of the HMD.

5.2 Registration of the RGB+D Sensor

Using the notation of [85], we will show how to recover the position of the Depth camera in the virtual world frame. Figure 5.3 outlines the relationships between the several frames discussed in this section. The important frames are H, the pose of the HMD,

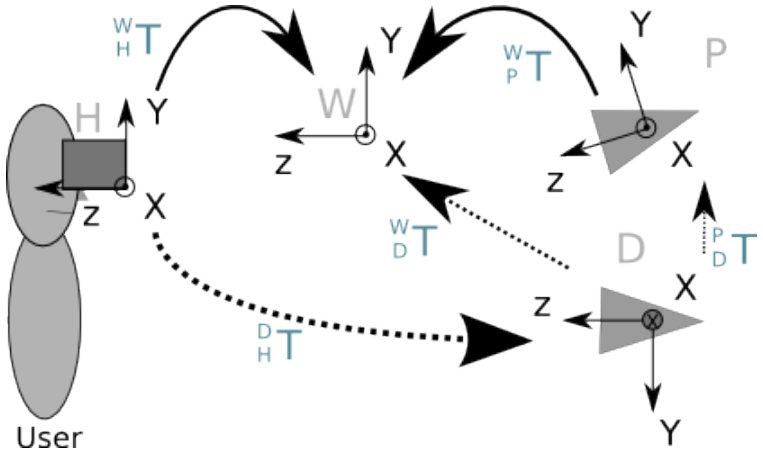


Figure 5.3: Illustration of the registration system, highlighting frames and transformations. Gray letters denote the names of the different frames: H, D, W, and P (See Section 5.2). Blue text denotes transformations. The arrows denote the direction of the transformation with the arrowhead indicating the resulting reference frame. Solid lines are transformations known *a priori*. Dash lines indicate transformation that need to be recovered.

P the pose of the HMD position camera, D the pose of the RGB+D sensor, and W an arbitrary world frame. There are also a series of already provided transformations, ${}^{W}_{P}T$ the transformation from the HMD position camera to the world frame and ${}^{W}_{H}T$ the transformation from the HMD to the world frame. And finally, there are the transformations that need to be recovered: ${}^{D}_{H}T$ the transformation from the HMD pose to the RGB+D sensor, ${}^{W}_{D}T$ the transformation from the RGB+D sensor to the world frame, and ${}^{P}_{D}T$ the transformation from the RGB+D sensor to the HMD position camera. Each frame transformation T is described by a transformation matrix storing rotation $R \in SO(3)$ and position $t \in \mathbb{R}^3$.

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \tag{5.1}$$

The first step is to recover the transformation from the RGB+D sensor to the HMD, ${}^{D}_{H}T$. However, the method in Section 5.1 doesn't necessarily measure the same HMD pose as the position camera, which needs to be corrected for. Instead, the transformation recovered by Section 5.1 is ${}^{D}_{S}T$, which is the sensed frame S of the HMD recovered w.r.t. the RGB+D sensor frame D. The transformation that registers the frame H that is

detected by the position camera w.r.t the sensed frame S on the HMD is referred to as ${}^{S}_{H}T$. In practice this was set to the identity but in the future can be determined empirically. With ${}^{S}_{H}T$, we can compute

$$ {}^{D}_{H}T = {}^{D}_{S}T\,{}^{S}_{H}T. \tag{5.2} $$

The HMD system is then able to obtain a measurement of the pose of the HMD that we wish to calibrate to called ${}^{P}_{H}T$. These two frames can be linked through a transformation as follows,

$$ {}^{P}_{H}T = {}^{P}_{D}T\,{}^{D}_{H}T. \tag{5.3} $$

Thus the transformation between the two sensors observing the HMD can be recovered using,

$$ {}^{P}_{D}T = {}^{P}_{H}T\,{}^{D}_{H}T^{-1}. \tag{5.4} $$

In the case of the transformation matrices used here, the inverse transformation is computed as the inverse of the transformation matrix [85].

While not strictly necessary, there is commonly a virtual world frame that the HMD position is transformed into, as in Figure 5.3. In this case, it is more practical to recover the RGB+D camera frame w.r.t. the virtual world frame W. Using the result from Equation (5.2) and the method presented in Section 5.1, we can recover the transformation to the world frame with

$$ {}^{W}_{D}T = {}^{W}_{H}T\,{}^{D}_{H}T^{-1}. \tag{5.5} $$

This transformation then allows us to transform 3D points from the RGB+D sensor into the virtual world frame.

5.3 Experimental Results

Experiments were performed using the Oculus Rift DK2 HMD and its corresponding IR-based position tracking camera. The RGB+D device used was an ASUS Xtion PRO sensor. It is capable of producing RGB and Depth frames at 320 X 240 resolution each at 60 frames per second. An example setup can be seen in Figure 5.4.

Our implementation is run on a desktop computer with an Intel® Xeon® CPU E3-1270 v3 processor running at 3.5GHz with 32GB RAM and a NVIDIA® Quadro® K2000

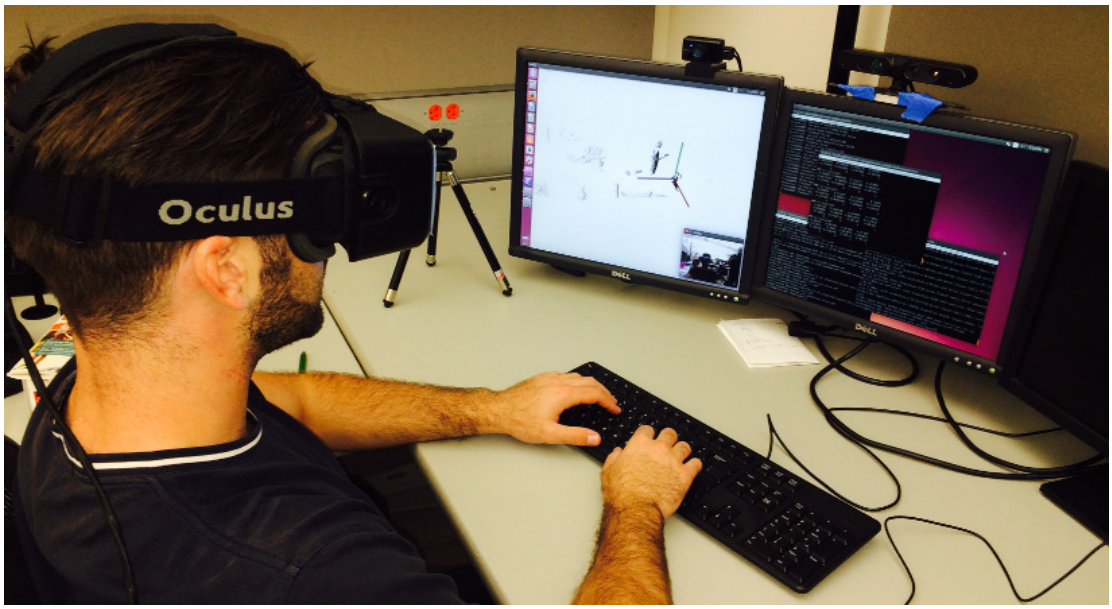


Figure 5.4: Physical example of the registration system.

graphics card. The average time for processing a single RGB and Depth pair with the proposed method is 84ms. This allows the possibility of real-time execution. However, this is not necessary as the sensor is presumed stationary. Once the transformation ${}^{W}_{D}T$ is recovered, it only needs to be stored so that it can be used to transform incoming depth images from the RGB+D sensor. The RGB camera on the RGB+D sensor was calibrated using the Caltech calibration toolbox [86] to recover the intrinsic matrix used for depth point projection. The depth images were coregistered with the RGB images making the RGB camera calibration appropriate.

5.3.1 Verification with Point-Tracking system

In order to verify both positional accuracy as well as rotational accuracy, testing was performed with a Vicon point-tracking system. This system is comprised of ten Vantage T16[1] cameras and is used to track IR-reflective markers. Even though the Oculus system also uses IR for position tracking there was no evidence of system interference.

[1] `https://www.vicon.com/products/camera-systems/vantage`

Figure 5.5: Tracking fixtures used for the point-tracking system.

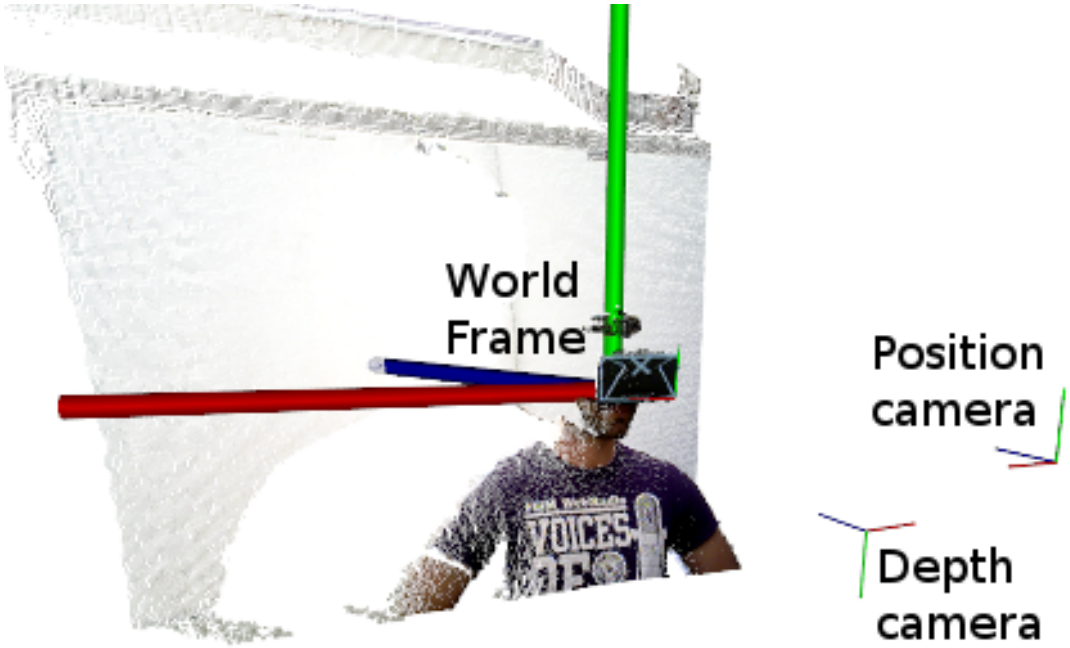


Figure 5.6: An example of the result from running the registration system. The frame of the HMD is being drawn as well as the frame of the RGB+D sensor and the position camera. The largest coordinate frame corresponds to the virtual world frame. The RGD+D sensor is correctly localized as being approximately 1m away.

Fixtures shown in Figure 5.5 were attached to both cameras and tracked as independent objects using the Tracker software provided by Vicon. Even though the tests are performed with stationary position and RGB+D cameras, tracking the objects allowed

for quick reconfiguration for each trial.

Several trials were performed to see how well the method performs for different situations. The result of one trial is depicted in Figure 5.6. The relative transformation ${}^{P}_{D}T$ between the two cameras is determined by the presented method and compared using the measurement from the point-tracking system, ${}^{\hat{P}}_{\hat{D}}T$, as ground truth. The point-tracking system recovers ${}^{\hat{P}}_{\hat{W}}T$ and ${}^{\hat{D}}_{\hat{W}}T$, in the point-tracking system world frame $\hat{W}$, which are related by,

$$ {}^{\hat{P}}_{\hat{D}}T = {}^{\hat{P}}_{\hat{W}}T \, {}^{\hat{D}}_{\hat{W}}T^{-1} \tag{5.6} $$

Both the measurements from the presented method $\left({}^{P}_{H}T, {}^{D}_{H}T\right)$ and the point-tracking system $\left({}^{\hat{P}}_{\hat{W}}T, {}^{\hat{D}}_{\hat{W}}T\right)$ are averaged independently over successive samples in time then combined using equation (5.4) and (5.6) for comparison. The rotation part of the transformation matrix is averaged by summing element-wise the quaternion representation of each rotation component of the measured transformations and normalizing [87].

The placement of the RGB+D sensor was modified across different trials and the configuration of each trial is presented in Table 5.1. Various different configurations were tried to explore the range of the system while using point-tracking as ground truth. The results for each of the trials is presented in Table 5.2. Offset error, ΔT, is defined as the Euclidean distance between the measured transformation and the ground truth. Since offset error might mask information about which dimension an error occurs in, the absolute difference for each dimension is reported as well.

Rotations are compared by decomposing the rotational component of the measured ${}^{P}_{D}T$ from the proposed method into $R_{XYZ}(\gamma, \beta, \alpha)$ fixed angles. The same is done for the observation from the point-tracking system. The absolute difference in these component angles is recorded. This is inspired by the Euclidean distance between Euler angles metric discussed in [88]. To remove ambiguity in decomposition the following restrictions were imposed: $\alpha, \gamma \in [-\pi, \pi); \beta \in [-\pi/2, \pi/2)$.

In general the rotational accuracy is within an acceptable limit with most trials having a rotation error of $< 8^{\circ}$ in any axis. The offset error could be improved with the worst magnitude being $>$ 14cm. In practice, while having this error, the intent of the proposed method is still accomplished. Figure 5.7 indicates this is a reasonable assumption as the error in alignment is imperceptible.

Several potential sources for error exist in both the proposed methodology as well as

Trial #	RGB-D Camera Position		
	Lateral Location	Vertical Location	Distance
1	50cm to right	Same Height	Same Distance
2	50cm to right	20cm Above	Same Distance
3	50cm to right	Same Height	25cm Behind
4	50cm to right	20cm Above	25cm Behind
5	50cm to left	Same Height	Same Distance
6	50cm to left	20cm Above	Same Distance
7	50cm to left	Same Height	25cm Behind
8	50cm to left	20cm Above	25cm Behind
9	Inline	20cm Above	Same Distance
10	Inline	20cm Above	25cm Behind

Table 5.1: Point-tracking verification test configurations. RGB-D camera position is relative to position camera. Perspective is from user facing cameras.

Figure 5.7: Stereo pair from the perspective of the HMD user of registered RGB+D data as a point cloud. During this example, the user is moving a paper cube and an insulated can holder. This gives an indication to the accuracy of registration and shows it is suitable for the proposed application. The gaps occur as a result of occlusions from the viewpoint of the RGB+D sensor. See supplemental material for a full video.

#	ΔT [m]	ΔX [m]	ΔY [m]	ΔZ [m]	$\Delta\alpha$ [°]	$\Delta\beta$ [°]	$\Delta\gamma$ [°]
1	0.0187	0.0027	0.0102	0.0155	2.1035	3.0524	4.4305
2	0.0276	0.0129	0.0059	0.0236	5.8999	5.0165	6.4910
3	0.0472	0.0288	0.0364	0.0085	9.4734	5.0327	5.1919
4	0.0461	0.0079	0.0398	0.0220	2.6004	4.6107	2.6127
5	0.1201	0.0311	0.0350	0.1106	1.5183	7.7690	5.7229
6	0.1276	0.0133	0.0787	0.0996	6.9229	7.5071	6.6108
7	0.1408	0.0662	0.0537	0.1121	4.8242	8.4543	6.3108
8	0.1195	0.0519	0.0495	0.0956	0.3570	8.6709	1.1453
9	0.0528	0.0250	0.0149	0.0441	5.4605	4.1343	3.9555
10	0.0786	0.0583	0.0258	0.0460	7.3932	5.4458	5.6639

Table 5.2: Results for each trial using the point-tracking system

the comparison for ground truth. The proposed methodology is limited by the resolution of each camera involved in the system in terms of being able to measure the (x,y) position of the HMD in the image plane. This accuracy is decreased as the HMD is moved away from any of the image sensors. In our experiments, the HMD was never more than 1m away from either the position camera or RGB+D sensor. The estimation of the intrinsic calibration parameters of the RGB camera can also effect measurements done by the proposed system as it relies on this for accurate 3D projections. The resolution and accuracy of the depth sensor can also effect the overall accuracy of the proposed method.

Verification using the point-tracking Vicon system also has limitations. Of course the same limitations with image resolution occurs with these systems as well. Locating the camera centers for the position and RGB+D cameras is prone to error. Without disassembling the cameras, these positions have to be approximated by placing fixtures near where the camera centers are likely to be. In light of these sources of error the measured errors are minimal.

5.4 Summary

This chapter provided a method for bringing together RGB+D sensors and HMD displays. The use of a registered RGB+D sensor with an HMD system can enable natural

interaction as well as provide the user with increased user presence in a virtual world. The following chapter addresses how this realized using a markerless interaction approach.

Potential improvements could be made to the methodology presented in this chapter. Using robust markers for localization such as ArUco markers [89] could improve reliability. The effective placement of such markers would need to be studied as they may occlude the HMD in such a way that its pose cannot be recovered from it's position system. An ideal methodology would require no modification to any of the devices involved. An alternative way for locating the faceplate of the HMD might me possible using advancements in object detection such as in [6].

Chapter 6

Markerless Interaction

This chapter discusses an approach for markerless interaction using an RGB+D sensor. Markerless interaction allows a subject to interact naturally with the virtual environment. Any sensing and computation done to facilitate natural interaction must be done quickly to minimize the amount of compensation the user has to afford for interaction. Recent advances in graphical processing units (GPUs) and highly parallel algorithms make this possible. Modern GPUs can have over a thousand processing cores and multiple gigabytes of on board memory enabling highly parallel tasks. Each core is optimized for parallel access to memory and arithmetic operations and unlike CPU cores performs poorly when branching. This architecture was initially designed for purely rendering purposes by allowing programmers to write per-pixel programs known as *shaders* to add increased detail to graphical rendering. Application programming interfaces (APIs) such as *CUDA*[1] and *OpenCL*[2] have opened this hardware up to general purpose computing.

The following sections detail how this markerless interaction can be made possible. Using an RGB+D sensor, 3D points from the observed scene can be recovered using the method discussed in Section 6.1. The information from successive RGB+D frames are used to infer 3D motion as discussed in Section 6.2. By having information on the 3D structure and motion in the scene it is possible to track trajectories over time as presented in Section 6.3. Tracking these trajectories provides input to a particle based physics engine discussed in Section 6.4.

[1] `http://www.nvidia.com/object/cuda_home_new.html`
[2] `https://www.khronos.org/opencl/`

6.1 Projective Geometry and RGB+D cameras

This section describes information regarding projective geometry and its use with RGB+D cameras. These cameras provide a color and depth image $D : \Omega \to \mathbb{R}$ at each time instance t over an image domain $\Omega = \{x, y\} \subset \mathbb{R}^2$. The color image components (red, green, blue) can be averaged at each pixel to derive an intensity image $I : \Omega \to \mathbb{R}$. For the purposes of this work, the RGB+D cameras are modeled as a pinhole cameras. Since only the depth camera is used for geometric reconstruction all of the remaining discussion in this section refers to that camera. The camera matrix P is a 3×4 matrix that maps a homogeneous 3D coordinate $\mathbf{x}_w = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ in the world frame to a 2D pixel $\mathbf{x}_i = \begin{bmatrix} x & y & 1 \end{bmatrix}^T$ in the image,

$$\mathbf{x}_i = P\mathbf{x}_w. \tag{6.1}$$

World frame refers to an arbitrary frame tat is used as the global frame of reference to which each device is calibrated. It is comprised of the 3×3 matrix K, known as the intrinsic matrix, and the extrinsic parameters of 3×3 rotation matrix R and 3×1 inhomogeneous translation vector t that transform the point $\mathbf{x}_w$ into the reference frame of the camera.

$$P = K \left[R \;\middle|\; t \right] \tag{6.2}$$

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{6.3}$$

where α_x, α_x are the focal length parameters, s is the skewness factor and x_0, y_0 are the camera center offsets. These parameters can be found experimentally using the method first discussed in Section 5.3. A thorough examination of camera models and calibration approaches can be found in [90].

The inverse of P is a 4×3 matrix given by the equation,

$$P^{-1} = \begin{bmatrix} R^{-1} & -Rt \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} K^{-1} \\ 0 \quad 0 \quad 1 \end{bmatrix}, \tag{6.4}$$

where $\mathbf{0}$ is a 3×1 zero vector. This can be used to map an image point (x, y) to homogeneous world point. However this requires knowing the depth Z for that point, which is provided by image D from an RGB+D camera. Therefor when recovering the world coordinate using Equation (6.4), Z should be used like so,

$$\mathbf{x}_i = \begin{bmatrix} xZ \\ yZ \\ Z \end{bmatrix}. \tag{6.5}$$

Applying this equation at every pixel in D yields a collection of 3D points O often referred to as a point cloud. In practice, not every pixel in D is a valid pixel. The reasons for this occurring include the depth sensed at the pixel is outside of the sensor's prescribed range and occlusions created specific to the sensor's design. These can be masked as $D_{x,y} = 0$.

6.2 Scene Flow

Scene flow is the 3D extension of optical flow, first introduced in [91]. Instead of describing the velocity of a pixel from frame to frame, it is the velocity of a 3D point from frame to frame. In traditional cameras this is impossible to compute without simplifying assumptions. Some works have presented methods for computing scene flow using stereo cameras [91, 92, 93], thus providing a constraint in which to recover changes in depth. RGB+D cameras provide this depth information with every frame, offloading potentially expensive stereo computation.

A simple way to recover scene flow using the image I and depth D information at every frame t from an RGB+D camera source is to compute the optical flow between frames I_t and I_{t+1} and use that mapping to recover the change in depth. This has been done in [51] however it is not the most accurate as the optical flow computation is not in any way constrained by the depth information provided. Jaimez *et al.* [94] present a real-time method for computing scene flow from pairs of RGB+D frames that does incorporate depth information. The algorithm attempts to determine the scene flow vector $\mathbf{s} = \begin{bmatrix} u & v & w \end{bmatrix}^T$, which is comprised of the optical flow $(u, v)$and range flow w, at each pixel between pairs of image and depth frames by solving the problem

$$\underset{\mathbf{s}}{\text{minimize}} \quad E_D(\mathbf{s}) + E_R(\mathbf{s}). \tag{6.6}$$

The first term

$$E_D(\mathbf{s}) = \int_\Omega |\varrho_I(\mathbf{s}, x, y)| + \mu(x, y)|\varrho_Z(\mathbf{s}, x, y)|dxdy \tag{6.7}$$

encourages solutions for $\mathbf{s}$ that maintain brightness consistency,

$$\varrho_I(\mathbf{s}, x, y) = I_0(x, y) - I_1(x + u, y + v) = 0, \tag{6.8}$$

and geometric consistency,

$$\varrho_Z(\mathbf{s}, x, y) = w - D_1(x + u, y + v) + D_0(x, y) = 0. \tag{6.9}$$

$$\mu(x, y) = \frac{\mu_0}{1 + k_\mu \left(\frac{\partial Z}{\partial x}^2 + \frac{\partial Z}{\partial y}^2 + \frac{\partial Z}{\partial t}^2 \right)}, \tag{6.10}$$

balances the contribution of the geometric term with the brightness consistency term, emphasizing the geometric consistency in areas with low depth gradients. The parameters μ_0 and k_μ are tunable. The second term

$$E_R(\mathbf{s}) = \lambda_t \int_\Omega \left| \left(r_x \frac{\partial u}{\partial x}, r_y \frac{\partial u}{\partial y} \right) \right| + \left| \left(r_x \frac{\partial v}{\partial x}, r_y \frac{\partial v}{\partial y} \right) \right| dxdy + \lambda_D \int_\Omega \left| \left(r_x \frac{\partial w}{\partial x}, r_y \frac{\partial w}{\partial y} \right) \right| dxdy \tag{6.11}$$

where

$$r_x = \frac{1}{\sqrt{\frac{\partial X}{\partial x}^2 + \frac{\partial Z}{\partial x}^2}}, r_y = \frac{1}{\sqrt{\frac{\partial Y}{\partial y}^2 + \frac{\partial Z}{\partial y}^2}} \tag{6.12}$$

regularizes the output $\mathbf{s}$ by considering the total variation while also respecting the geometry of the scene by scaling using r_x, r_y which favors close points as opposed to distant ones. The data term E_D is non-convex, leading the authors of [94] to adopt a coarse-to-fine scheme where an image pyramid is built and solutions from lower levels of the pyramid are upscaled and used in linearized versions of $\varrho_I(\mathbf{s}, x, y)$ and $\varrho_Z(\mathbf{s}, x, y)$. The solution to Equation (6.6) is then computed using an iterative primal-dual solver [95] whose pixel-wise updates are amenable to a parallel computation on a GPU.

Implementation of this method (PD-Flow) is compared against GPU based optical flow methods provided by OpenCV [83]. Per-frame run-times for alternative methods and the proposed method are shown in Table 6.1. While it is not the fastest method presented, the method does incorporate depth information into its optimization for scene flow leading to more accurate results as depth can disambiguate issues with occlusion. Scene flow vectors at each pixel provide important information for longer term tracking.

Method	Per-frame run-time (ms)
Farneback	22.534
Brox	99.585
PD-Flow	80.264

Table 6.1: Scene flow implementation runtimes

6.3 Trajectory Tracking in RGB+D cameras

Using the method described previously for computing scene flow and the following function,

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} \frac{Z}{\alpha_x} & 0 & \frac{X}{Z} \\ 0 & \frac{Z}{\alpha_y} & \frac{Y}{Z} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \tag{6.13}$$

applied at each pixel, gives us a tensor $F \in \mathbb{R}^{m \times n \times 3}$ that describes the estimated velocity $F^t_{x,y} = (U, V, W)$ from each projected pixel in time t to $t + 1$.

In order to understand the motion of points through successive frames we adopt an approach similar to [28] extended to scene flow. This is desired because it produces features amenable to activity classification as discussed in Chapter 3 as well as allowing for correct progression of points from frame to frame for the purpose of input for 3D interaction.

Each tracker $T_i = \{\mathbf{x}_j\}_{j=1}^{L}$ where $i = 1, ..., k$ is initialized at a regular interval horizontally and vertically along the image. Each tracked point $T_i^j \in \mathbb{R}^3$ is mapped back to a pixel coordinate (x, y) using Equation (6.2). Median filtering is applied to current pixel and the 8-connected neighbors around $F^t_{x,y}$ to produce an estimate of the motion (U, V, W) to apply to that point in the next frame. Filtering is done to reduce noise from the scene flow computation. That point is then progressed given the filtered estimate of (U, V, W) to determine T_i^{j+1}. Tracked points that were not matched, such as in the case when $D_{x,y} = 0$, are discarded immediately. A dense sampling pass then occurs again creating new tracks at regularly sampled intervals that do not have a track in that frame associated with them. A given track is updated for a fixed lifetime L before it is then removed from tracking. As with [28], this is done to improve the correctness of tracks as accuracy is likely to drift over time. Complete tracks of length L can be stored for offline analysis.

Figure 6.1: An example of 3D trajectory tracking. Progress of the tracks proceed from dark green to light green. Blue points represent the end of the tracks. The subject is moving their arm downward.

This approach yields a set of trajectories over different time intervals that describe more complex motions of the observed point cloud O than simply using scene/optical flow. An example of the result of this algorithm can be seen in Figure 6.1. The endpoint positions of the current tracks and their respective velocities are then provided as input at each time step for interaction with virtual objects.

6.4 Interaction using Scene Flow and Trajectory Tracking

Being able to track points from to O^t to O^{t+1} makes it possible to assign a velocity to each point. This opens up the possibility to potentially track thousands of points at each time frame. Directly using these points as entities inside a physics simulation can lead to a large amount of entity to entity interaction that needs to be performed efficiently.

An efficient way to handle particle-based physics using GPUs is presented in [96] and implemented in the FLeX engine [97]. Their approach is a position-based dynamics method that accepts particle positions, velocities, masses and constraints C_i as inputs and simulates the dynamics for a fixed timestep by solving the optimization problem

$$\begin{aligned} \underset{\Delta x}{\text{minimize}} \quad & \frac{1}{2}\Delta x^T M \Delta x \\ \text{subject to} \quad & C_i(x + \Delta x), \quad i = 1, ..., n, \end{aligned} \tag{6.14}$$

where $M = diag(m_1, ..., m_p)$ is the mass matrix, $x \subset \mathbb{R}^{p\times 3}$ are p particle positions, and $\Delta x \subset \mathbb{R}^{p\times 3}$ are the corresponding particle displacements. The constraints describe different relationships between particles that must be obeyed throughout a simulation step. They are usually non-linear and non-convex functions preventing a closed form for solution to Equation (6.14). Instead the constraints are locally linearized creating a quadratic program with linear constraints that is solved using successive over relaxation. The displacements are solved for in two stages. First, collisions are determined between particles in parallel using an efficient-hash grid methodology [98]. This imposes collision constraints that are then solved for. Each particle is assigned to a single contact group. A contact group is a set of particles that have the same interaction characteristics. The group can either allow for self collisions between particles in the same group or not. For instance, the particles of each rigid object are declared as their own group and are

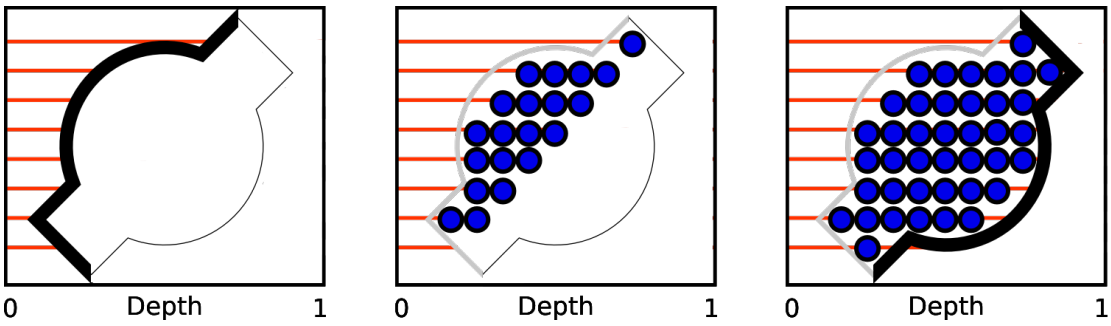
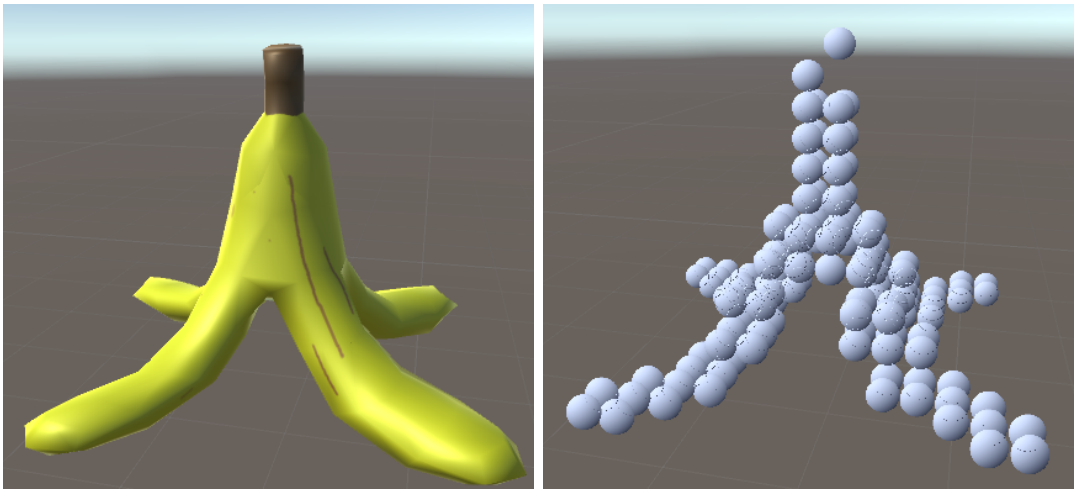


Figure 6.2: An example of depth peeling stripping away successive layers with each pass. After every odd layer, particles (blue) are populated along the rays (red). The leftmost surfaces are represented as thick black lines. The hidden surfaces are depicted as thin black lines and gray lines represent peeled away surfaces.

not self-colliding. This has the benefit of pruning unnecessary collision and dynamics computations. After which the remaining constraints are solved for and this can be done in either a contact group or per particle fashion in parallel.

Solid objects with a structure more complex and sizes larger than a particle are modeled as a collection of particles with rigid body constraints keeping the structure of the object enforced. This complex shape is described using a tri-mesh or a list of 3D vertices and a list of triplets connecting vertices to denote triangles. A watertight tri-mesh is required for this method. Given a tri-mesh describing the size and shape of an object it is possible to transform that object into a collection of particles of arbitrary size using a method referred to as depth peeling. Depth peeling is the process of ray casting from an arbitrary direction to test tri-mesh intersection (See Figure 6.2). In each ray cast pass, the tri-mesh touched by the ray cast is removed or peeled away. In the next iteration the next level of the tri-mesh, which is exposed, is removed [99]. Particles are populated at regular intervals along the ray at each alternate peeling (e.g. the first intersection begins the process of adding particles then the second intersection terminates until the third intersection etc.).

The result is a particle representation of the tri-mesh (See Figure 6.3b). A drawback of this approach for modeling rigid objects inside of the physics engine as particles of fixed sizes is it limits how large an object can be while remaining tractable. Larger objects require even more particles and reducing the resolution of depth peeling can lead to a situation known as *tunneling* where objects interpenetrate each other. As



(a) (b)

Figure 6.3: The result of depth peeling the object depicted in (a) is shown in (b).

a result large static objects such as walls are represented using a traditional tri-mesh description and modeled in the physics engine as static constraints.

Individual tracked points are modeled as particles in the physics engine. The radius of the particles should be set to span half the minimum distance between two points in a point cloud O. In practice this was set to $r = 3$cm. This means that hand-held virtual objects are still able to retain tractability being representable in hundreds of particles each.

6.5 Summary

This chapter presented a method for markerless interaction using RGB+D data. It was made possible by selecting approaches amenable for a high degree of parallelization realized on a GPU. A key aspect effectively working with a GPU for computation is minimizing the communication load between the CPU and the GPU. Further improvements to performance can be made by keeping the results of subsequent steps local to the GPU.

The speed of the capture sensor, scene flow calculation, and physics update can all affect the interactivity of this approach and are coupled together. The lower the

sensor framerate, the slower the motions of the subject will need to be in order to be captured by the scene flow algorithm. Large displacements sensed through a low frame rate can mean that particles which should interact with objects instead jump past what the subject intended to interact with.

The ability to interact with virtual objects naturally enables a more believable experience and facilitates the experiment discussed in the following chapter. What remains is creating proper scenarios that elicit symptoms and provide measurable diagnostics. These diagnostics can then be used as part of an overall understanding of a subject.

Chapter 7

Study In Immersive Environments

This chapter describes how the findings and methodologies in the previous chapters are brought together to provide an end-to-end immersive system for mental health assessment. Having an immersive system allows for a controlled environment that is repeatable, reconfigurable and allows the user to naturally interact with the environment. This system also has the potential to allow for a greater variety of scenarios when studying subjects. In contrast, performing an ecologically based assessment like the one presented in Chapter 4 required furnishing various location settings with equipment in order to examine different conditions. By using a virtual environment the system is only limited by the computational power of the computer rendering the scene and processing the simulation.

An overview of the entire system is given in Section 7.1. The diagnostic measures collected by the system and their potential clinical relevance are discussed in Section 7.2. An example scenario and protocol for executing that scenario are given in Section 7.3. Measurements from each of the trial runs were collected for a single subject performing different behaviors showcasing the measurement qualities of the system. The results from these trials are presented in Section 7.4. A discussion on the lessons learned for designing future scenarios is presented in Section 7.5.

7.1 System Overview

The hardware of the system consists of an HMD with pose tracking capability and an RGB+D sensor both connected to a computer. The subject is seated in an unobstructed space that allows the user to be observed by the necessary sensors (see Figure 7.1). This system assumes that no other occupants are in the observed range. This assumption is important because it simplifies locating the subject and also safely allows the subject to move without obstruction.

Scenarios are designed using an authoring tool such as the Unity3D[1] editor. Typically these tools are associated with a rendering engine that provides a means for rendering and controlling the virtual environment. Numerous user friendly tools for authoring environments have been created. NeuroVR[2] is one example of these tools particular to the psychiatric domain. While being easier to design environments, the tool lacked extensibility in being able to collect additional sources of measurement [66].

The scenario is then presented to the user via the HMD and they are able to interact with objects in the scenario. The RGB+D sensor is calibrated using the methodology discussed in Chapter 5. The methodology for resolving interaction with virtual object displayed in the scenario is discussed in Chapter 6.

These scenarios can vary up to the implementation and purpose for the disease and symptoms examined. One possibility would be to create a virtual handwashing station similar to the one presented in Chapter 4. As a stepping stone to such a scenario the example scenario implemented looks at cleaning compulsions with relation to larger more easy to simulate rigid objects instead of water. Another potential scenario would be to have an arrangement task inside of the virtual environment. This has the advantage that objects are already tracked as part of the simulation and would not require a visual tracking methodology to be implemented. Example diagnostic measures are presented in the following section. An example scenario is discussed further in Section 7.3.

The general processing pipeline for display and interaction remains fixed for every scenario (see Figure 7.2). After initialization, the system continuously captures RGB+D frames. Each pair of frames is used to compute scene flow. Only pixels corresponding to a specified depth interval are retained. This has the benefit of trivially segmenting out

[1] https://unity3d.com/
[2] http://www.neurovr2.org/

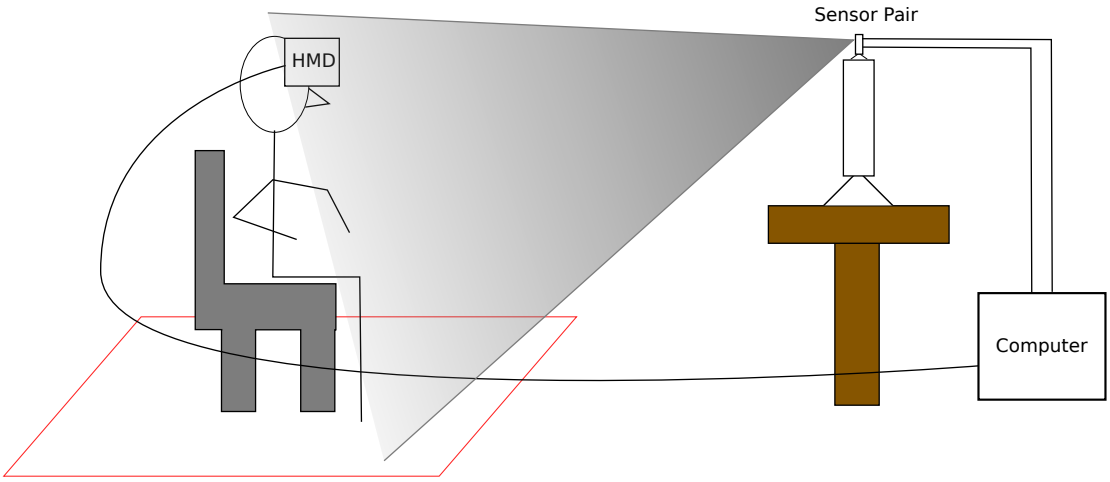


Figure 7.1: Diagram of the equipment setup for the proposed system. The area around the user should be unobstructed (as expressed by the red boundary.) The user should also be situated such that they are in the view frustum of the camera.

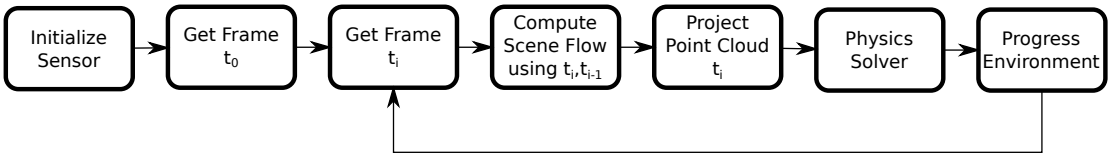


Figure 7.2: VR System Processing pipeline.

the subject from the environment observed by the RGB+D sensor. The latest frame is used for input in the point cloud computation. The retained 3D points and their corresponding scene flow vectors, as associated by projection onto the image plane, are input into the physics solver. This solver also takes input from the state of the particle representation of the objects employed in the particular scenario. As a result of the physics solver, the objects in the scenario are progressed in time. While the scenario executes the subject is monitored for various diagnostic measures as discussed in the following section.

7.2 Diagnostic Measures

The RGB+D sensor and HMD in use by the subject can provide a great amount of information about how they are interacting and reacting to the environment. In previous

psychiatric literature [17, 61, 63, 64], information about the head pose of the subject was of interest particularly in relation to the other tasks that were being performed as part of the scenario. As an improvement on these works, we propose maintaining statistics about the subject over time, as their reaction will likely evolve as the scenario progresses. This *in vivo* understanding has been desired by other researchers and lamented as a deficiency in earlier platforms [66].

7.2.1 Head Pose Tracking

Head pose tracking is necessary for VR delivered by an HMD to the user's eyes to function. It allows for the correct projection of the 3D virtual world. This information can also be used for diagnostic purposes. What follows are proposed measures that can be derived from the head pose information situation in the world frame.

Head Pose Attention measure The position (X_t, Y_t, Z_t) and orientation $(\alpha_t, \beta_t, \gamma_t)$ of the pose at any time instant t during the scenario provides a coarse understanding of the user's attention. The object of interest will be in the field of view of the subject. Analyzing the change in position and orientation, of the worn HMD, over time can provide a measure of how long the user fixates on any one object. This can be extended to categories of objects as it can provide an indication of preference or aversion to that category.

Head Pose History measures The history of the head pose throughout the time of the session is an important measure. It can provide an indication of how active the subject is during the session by gauging how their head moves around in their observation of the environment. Subjects may have a revulsion to some objects, which may manifest itself as sudden jerks in head pose. The following realizations of these measures are presented:

1. Total standard deviation in position, orientation, velocity, and angular velocity;
2. Sliding-window history of the standard deviations over time.

Total standard deviation in position and orientation over time characterizes how much the subject moved around during the the observed session. Total standard deviation in

velocity characterizes how varied the head motions are during the session. Providing a time-series of these measures computed over an interval in a sliding-window can provide insight into how these motions evolve over the course of a session. Selection of the size of the window varies the nature of what is examined. If the window is short, every variation will be captured but general trends may not be observed. Choosing too large of a window and transitions between low and high standard deviation could be missed.

7.2.2 Scene Flow Tracking

In addition to tracking the head pose, the motion of the subject's body in general is tracked via scene flow, as described in Section 6.3. This is important for capturing the level of activity and kinds of activity performed by the subject during the scene. One possibility is to encode the scene flow tracking data in a similar way as described in Section 3.1.2 for activity classification. This is the ideal use for this information as it brings a higher level understanding to the observed motion. This requires a data collection effort which is left as future work.

The scene flow trajectory data can be used to evaluate level of activity and how the subject moved in the scene. These trajectories, of which there can be thousands per recording, can be encoded for "at a glance" evaluation of activity. The total displacement of each scene flow trajectory, can be spatially binned, for a given time interval in an accumulation matrix A. Total displacement is computed by summing the lengths along a scene flow trajectory. First, the subject is localized by projecting all of the points along each trajectory in the time interval into the image plane and taking the maximal extents. A grid is then defined on this image region to summarize motion over the given time interval and location. Each cell of the grid is associated with an element of A. The total displacement of each trajectory is then accumulated into the appropriate cell in A. This occurs for every time interval. The resulting images $A_1, ..., A_k$ are then normalized to the dynamic range of all of the accumulating images. The result is a representation of the motion of the subject summarized over arbitrary time intervals. This allows for quick examination of the amount and location of motion during the session.

Computing the average total displacement over the entire observed sequence can also provide an assessment of how active the subject was. This can be used in conjunction with the head pose measure as the subject may have had their head still for the

entire session but the rest of their body did move around. Before averaging, low total displacement tracks should be removed as they likely exist due to noise in observation as opposed to actual movement by the subject.

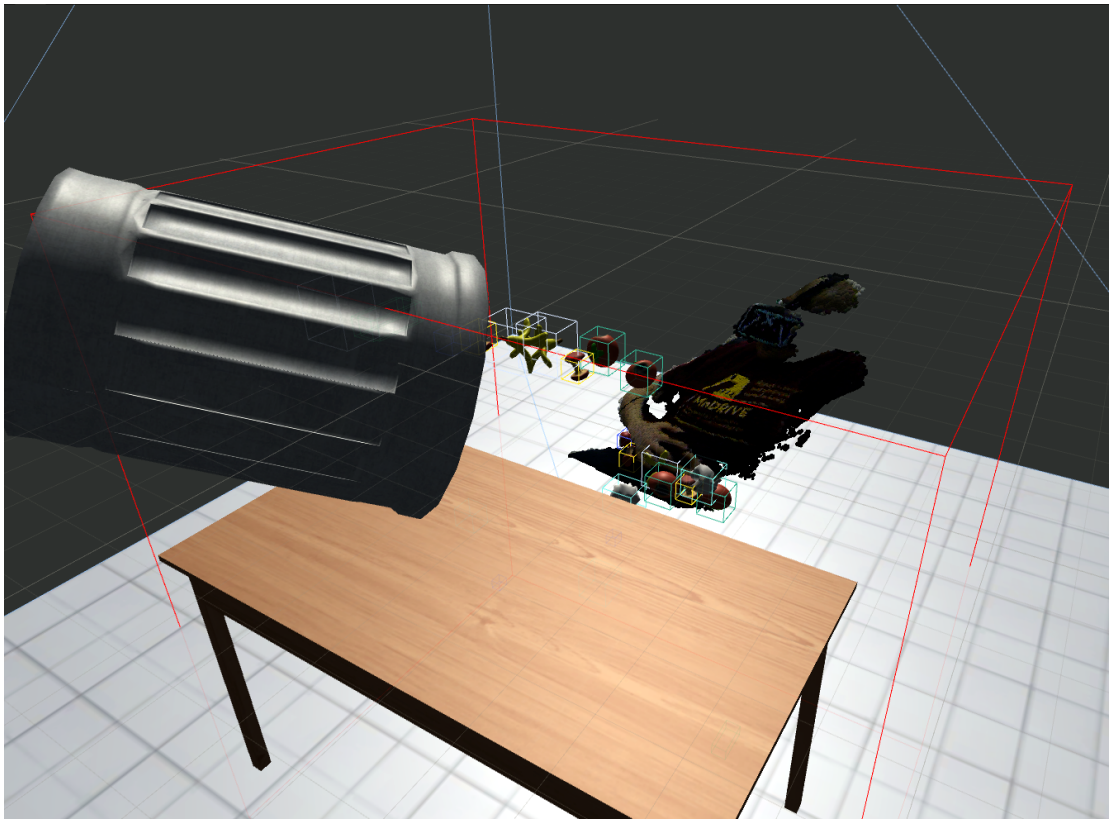


Figure 7.3: An example of the scene layout of the scenario's VR environment.

7.3 Scenario and Protocol

The proposed scenario has the subject sitting in a moderately sized room depicted in Figure 7.3. In that scenario they are able to view a point-cloud representation of themselves as seen in the Figure 7.4. This gives the subject the ability to reconcile their actual body movements with regards to the VR world. Each session should be run for five minutes to ensure that trends in behavior can be found. Throughout the five minute scenario, objects are projected towards the individual causing most of them to

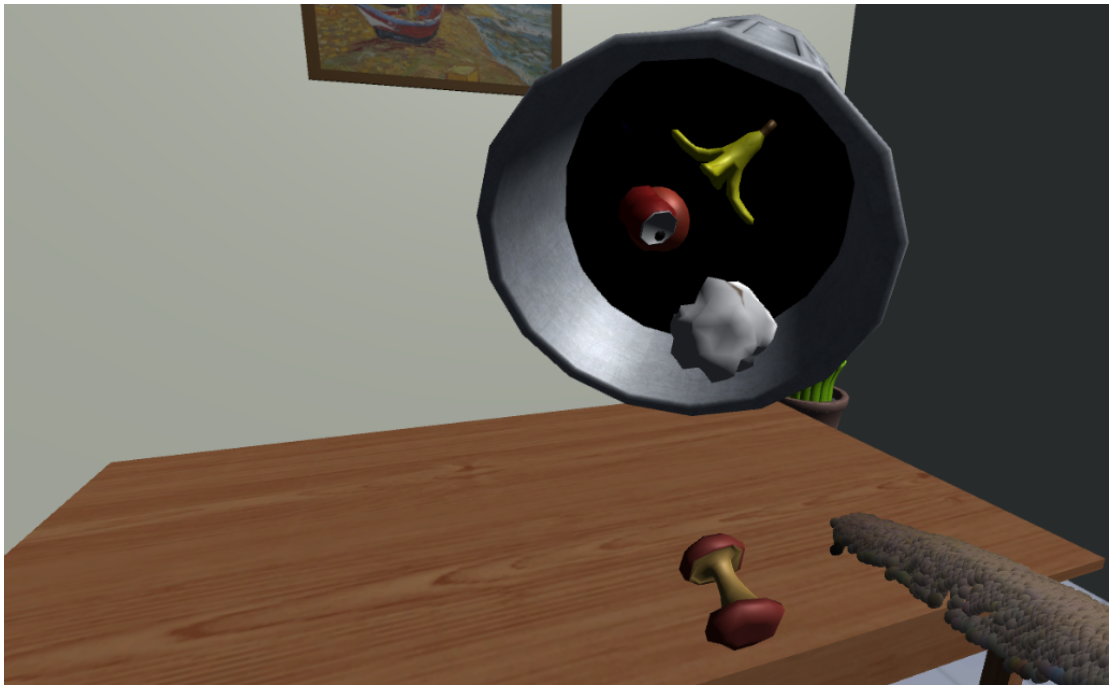
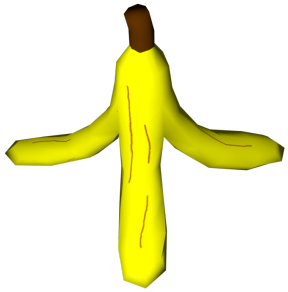


Figure 7.4: An example of the user's view in the proposed scenario's VR environment. The point cloud shows the points on the user that are tracked and included as part of the interaction of the system.

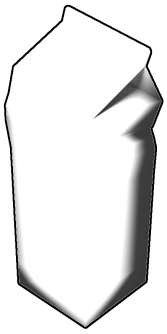
rest against the point cloud representation of the subject. Figure 7.6 depicts the virtual objects used in the proposed scenario. The subject is asked to try and clean away the objects from their body. An example of a subject clearing away objects can be observed in Figure 7.5. However more objects will continue to be projected towards the subject for the remainder of the scenario.

Figure 7.5: An example of a subject clearing off a waste paper object. The red box highlights the object touched by the user.

(a) Apple Core



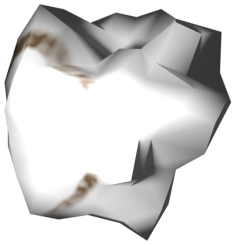
(b) Banana Peel



(c) Milk Carton

(d) Crumpled Pop Can

(e) Rotten Tomato



(f) Waste Paper

Figure 7.6: Interactive Objects Gallery.

This example scenario's purpose is to highlight the potential of this methodology to study behavior relating to cleaning compulsions. The objects selected for inclusion, depicted in Figure 7.6, were meant to represent examples of discarded items. It is expected that a normative subject would be indifferent to the objects as they approach in the virtual environment. However someone who suffers from a cleaning compulsion might be more active in trying to remove the mounting virtual objects from their person. Not all of the objects may have the same inciting effect which is why it may be important to track which objects the subject tends to focus on.

7.4 Experiment

The following experiment demonstrates the ability of the system to characterize the movement of the subject. This is important as it gauges how the subject reacts to the controlled immersive environment. The controlled environment allows for comparison between trial runs.

7.4.1 Setup

The example scenario was implemented using Unity3D version 5.4.2f2. The simulation was executed on a Windows 10 PC with an Intel® i7-6700K@4GHz processor with 32GB RAM and a NVIDIA® M5000 GPU. An ASUS Xtion PRO RGB+D sensor and Oculus Rift DK2 HMD were used for sensing the subject and displaying the virtual world to the subject respectively. Five trials were performed observing a single subject using the system.

The motions acted in each of the trials are described in the Table 7.1. In each trial the subject is tasked with cleaning the virtual objects flying towards them. If the subject does not clean away the objects, they can come to rest on the subject. Most of the cleaning occurs on the lower half of the subject's body.

7.4.2 Results

Each of these trials is meant to display the different types of motions that can be captured from observing the pose of the HMD and the motion of the subject during a session. For instance, Trial #1 and Trial #2 have the subject performing different

Trial #	Description
1	Head nodding up and down while trying to complete task.
2	Head shaking left to right while trying to complete task.
3	Staying still and not trying to complete task.
4	High level of activity while trying to complete task.
5	A typical attempt to try and complete task.

Table 7.1: Descriptions of the behavior exhibited during each of the trials.

	Trial #				
Measure	1	2	3	4	5
X	00.0124	00.0752	00.0064	00.0203	00.0715
Y	00.0603	00.0347	00.0215	00.0407	00.0516
Z	00.0254	00.0560	00.0119	00.0256	00.0327
α	24.0516	07.2349	08.7961	10.9967	15.8274
β	03.5415	39.1323	02.6964	06.7433	15.7990
γ	02.9255	04.1224	01.4549	03.4291	04.5316
$\dot{X}$	00.0003	00.0017	00.0001	00.0003	00.0005
$\dot{Y}$	00.0015	00.0003	00.0002	00.0005	00.0007
$\dot{Z}$	00.0006	00.0012	00.0001	00.0003	00.0003
$\dot{\alpha}$	00.6189	00.1159	00.0664	00.1682	00.2292
$\dot{\beta}$	00.0651	00.8930	00.0345	00.1331	00.2185
$\dot{\gamma}$	00.0726	00.1399	00.0124	00.0689	00.0923

Table 7.2: Standard deviation in head pose and velocity over five trials.

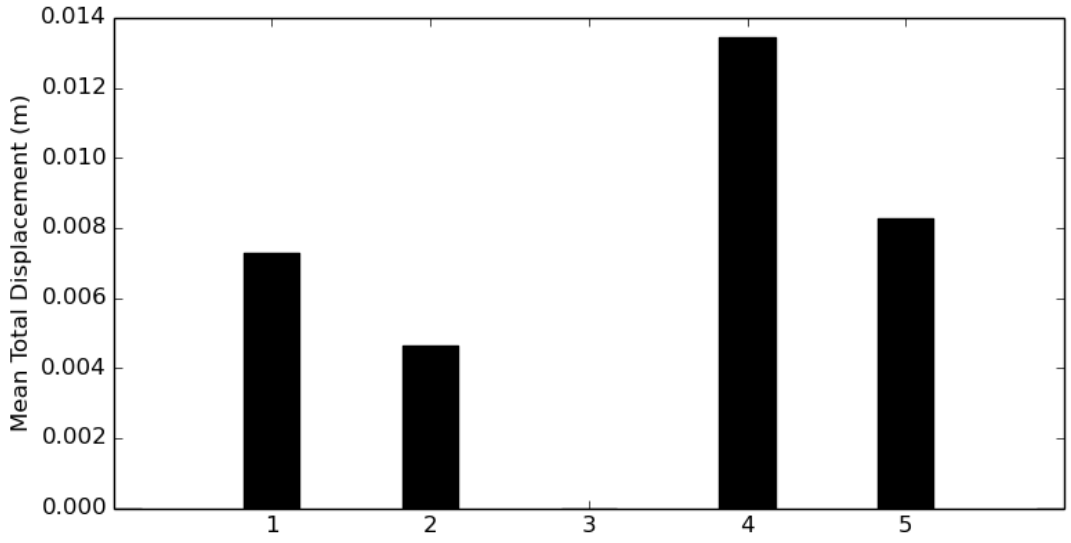


Figure 7.7: Mean Total Displacement of scene flow trajectories for each trial.

types of head movement during the scenario. Trials #3-5 have the subject performing different levels of activity. The total standard deviation statistics for each of the five runs is shown in the Table 7.2. Here it is shown that the dominant head motion for Trial #1 and Trial #2 is recovered. It is also possible to distinguish low levels of activity by the subject. In Trial #3 where the subject is performing the least amount of motion, the lowest standard deviation is reported for all measures. However the level of activity in the subject cannot really be distinguished between Trial #4 and Trial #5 based on head motion alone. This difference in activity is instead distinguished by comparing mean scene flow trajectory total displacement scores as shown in Figure 7.7. Scene flow trajectories with a total displacement < 0.006 were discarded before computing the mean. This measure also captures the comparatively low activity shown in Trial #3. Figure 7.8 and Figure 7.9 show the time-series plots of the head pose statistics for Trial #1 using a sliding window of 30 frames. Note the change in the standard deviation of α over time as the subject has to rest from head nodding. By having this time-series information it can highlight changes in the dominant head motion.

Figure 7.12 shows the amount of time the subject focused on each object during Trial #5. The subject intended to focus on the tomato objects to demonstrate the

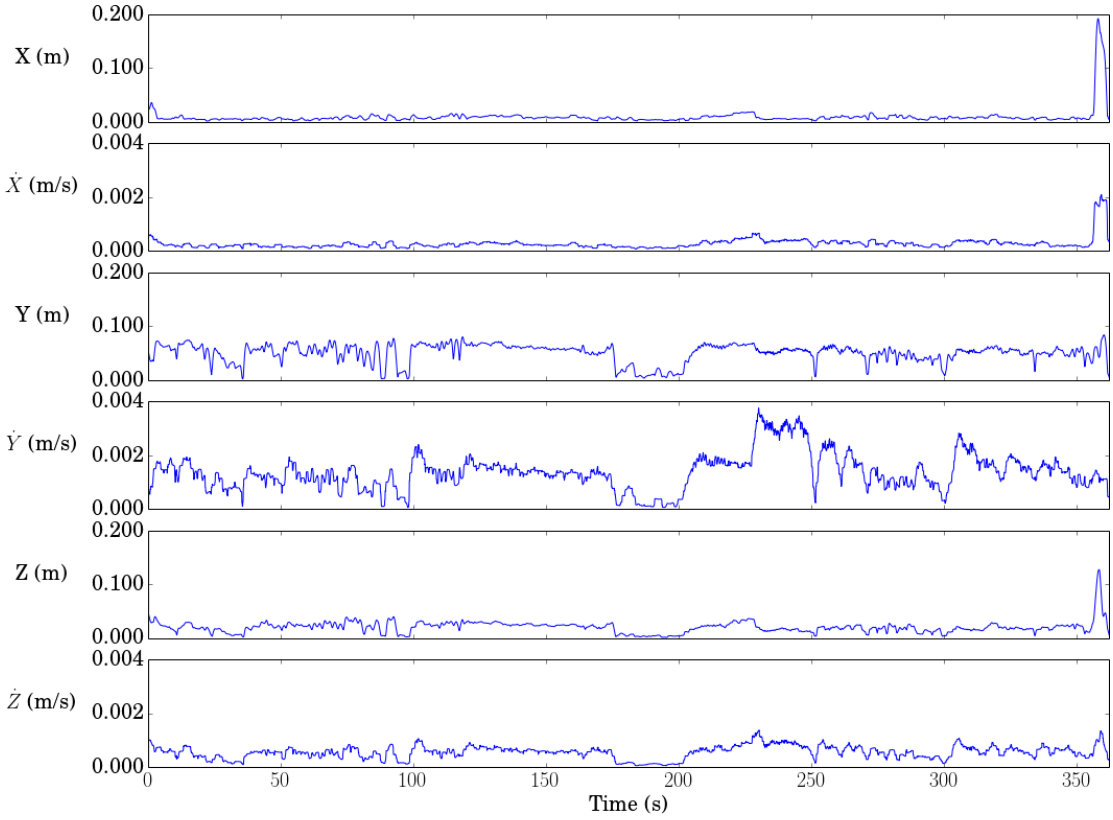


Figure 7.8: Time series of statistics recorded from the VR system during Trial #1 of the scenario.

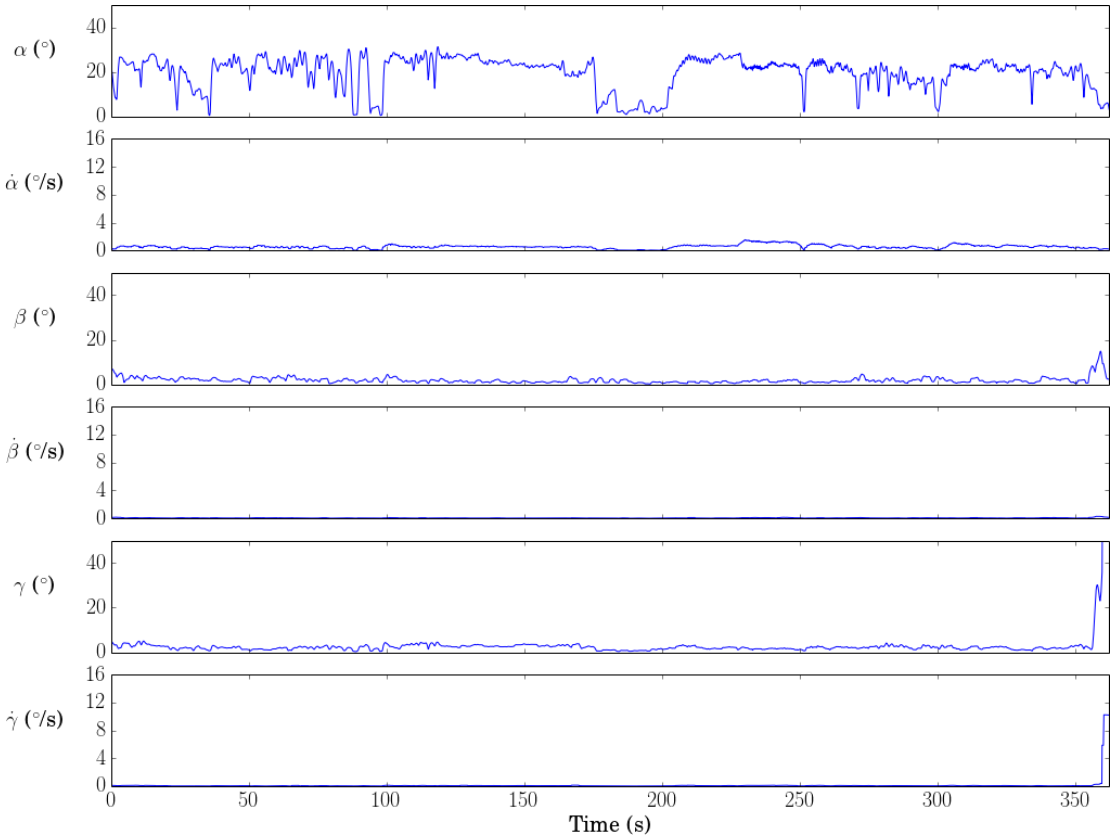


Figure 7.9: Time series of statistics recorded from the VR system during Trial #1 of the scenario.

effectiveness of the system in capturing focus of attention. The remaining object categories received less and approximately equal attention. With a measure like this it may be important to query the subject after they have completed the session to determine if their fixation to a particular object category has a motivation.

Figure 7.10 shows a summarization of the scene flow tracking result using the technique discussed in Section 7.2.2 during Trial #1 where the subject's head is nodding up and down as they attempt to complete the task. These images offer an "at a glance" assessment of the movement of the subject during the scenario. The primary motion exhibited in Figure 7.10 comes from the subject's head nodding and their arms deflecting objects in the virtual environment. It can be observed in frames 1351–1951 that most of this motion came from head nodding. This occurs again from 2251–2701. In the other intervals the activity is mixed between arm and head motion with the relative intensity encoded in the image. At the end of the session the subject needed to approach the computer to end the session. This explains the high degree of motion seen in 3901–4010, where the whole body of the subject moves towards the RGB+D sensor.

As a contrast, the same summarization was performed for Trial #4 in Figure 7.11. A majority of the motion performed by the subject during this trial is in their arms trying to clean away objects. The similar images suggest this motion is similar throughout the entire session. Unlike in Trial #1, barely any head motion occurs at the same intensity as the subject's arms. Executing these trials provided some insight into future scenarios.

7.5 On Designing Scenarios

A key challenge in designing scenarios for use with this system is one that is true for all virtual environments. That is the creation of 3D models and textures for objects that populate the environment as well as the environment itself. Fortunately there are repositories of available models on the Internet that can supplement in the construction of an environment. A remaining challenge is ensuring that these objects have the same relative scaling to each other. This is a mathematically trivial operation but it still requires work on behalf of the scenario implementer. Simply by changing the environment and the object used in interaction can alter the behavior of subjects and elucidate differences.

The task required for completion by the subject should take advantage of the sensing modalities available. For instance, in the default configuration of the proposed system, interaction may only take place in front of the RGB+D sensor. If the subject faces away from the sensor they will likely be occluding the view of their hands making interaction impossible. Furthermore it is not straight forward to signal to the subject in VR to stimuli behind them. Some clues may come from 3D audio or from peripheral vision. It is for these reasons that the activity should be available in front of the user.

Creating a new task for subjects to perform requires some consideration. The task that is required of the user should be simple and intuitive. This means that it should be as close to how the subject would naturally interact with real world objects. Even though the proposed methodology is designed to enable intuitive motion the subject will likely be new to the system. In the proposed scenario this was accomplished by having the task be swiping away objects. This way the user would not have to be concerned with understanding how to precisely interact with the objects allowing them to behave freely.

7.6 Summary

This chapter showed how the presented methodologies in this work can be brought together for mental health assessment. They extend upon previous approaches in several ways. They demonstrate the possibility to not only measure total movement of the head pose but also measures that vary in time which can characterize the subject's behavior over time. This is further augmented by the scene flow tracking based measures which give a measure of the total body activity of the subject. Measuring activity (or motion) serves as a indicator of engagement or reaction to eliciting stimulus.

Incorporation of 3D audio was only discussed briefly however its addition can certainly add to the level of immersion. Commercially available headphone systems exist for presenting 3D audio. Since all of the components are calibrated relative to the world frame and the pose of the subject's ears can be inferred using the pose of the HMD.

The best validation for the proposed approach would be to evaluate the system and a scenario in a study incorporating both a normative and affected population. Performing this validation would provide information on how the measures vary between groups

making modifications to the virtual environment and seeing how that changes behavior is an important question for further investigation.

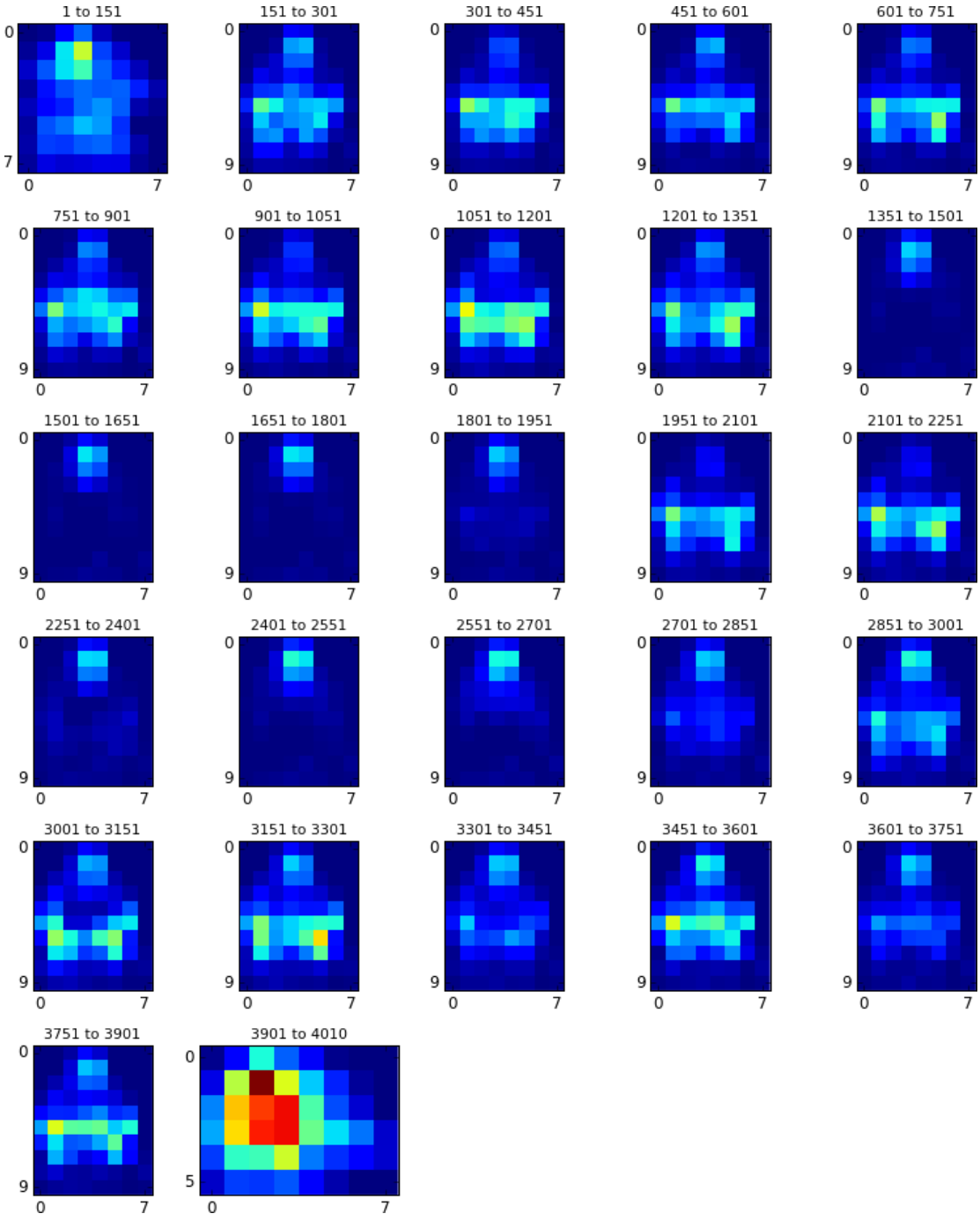


Figure 7.10: Scene flow tracking summary over 150 frame intervals from Trial #1. The amount of motion in a cell varies from low (blue) to high (red).

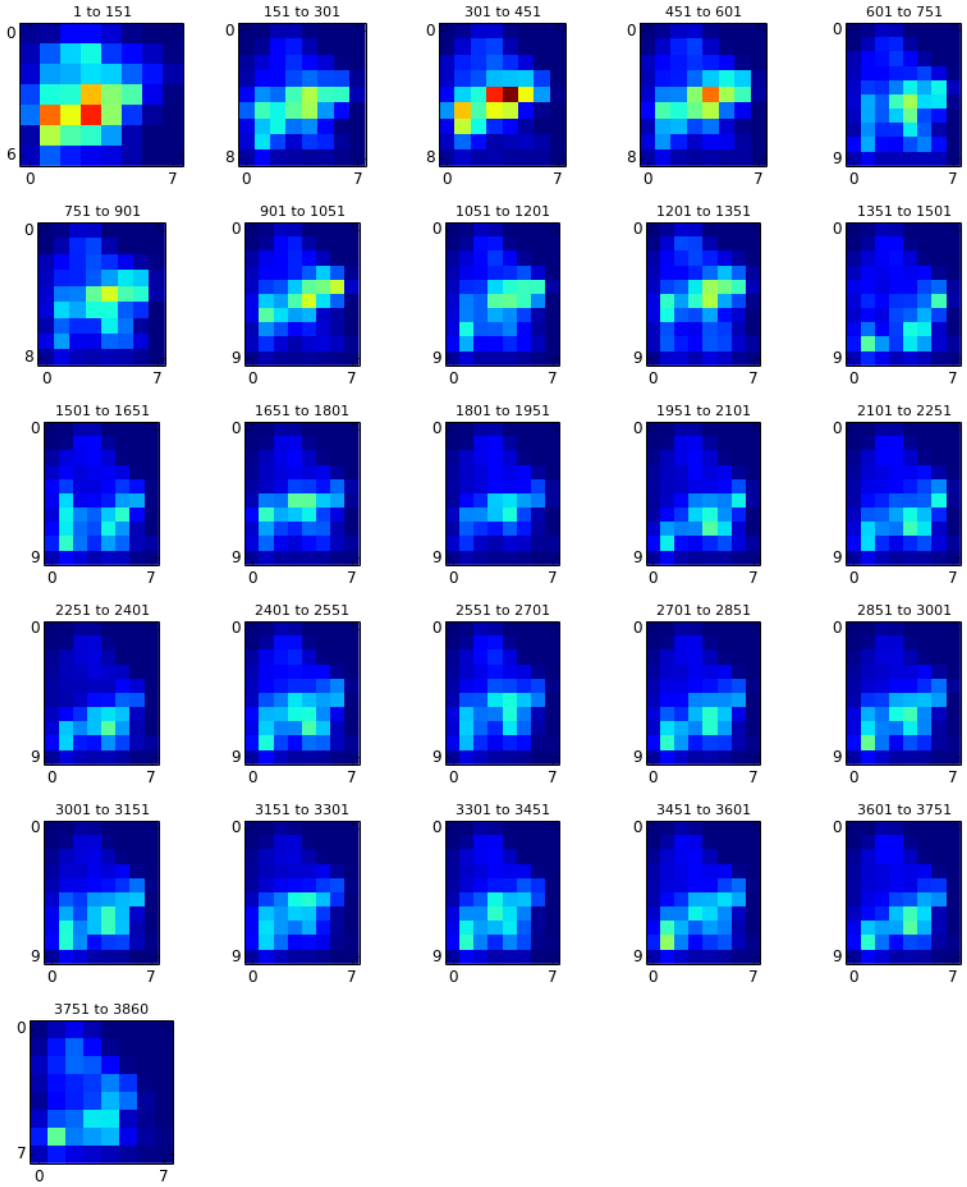


Figure 7.11: Scene flow tracking summary over 150 frame intervals from Trial #4. The amount of motion in a cell varies from low (blue) to high (red).

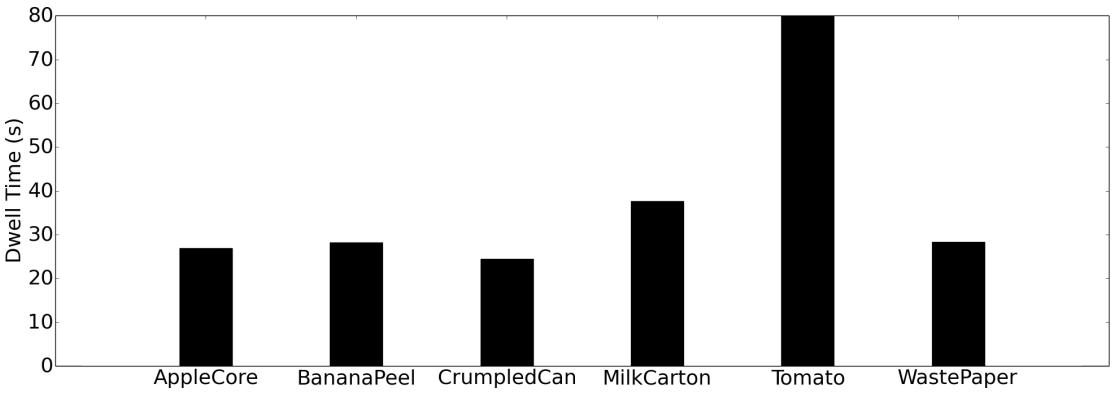


Figure 7.12: Time spent focusing attention on each type of object during Trial #5.

Chapter 8

Conclusion

This thesis investigated the use of computer vision and immersive environments in the application domain of mental health assessment. Computer vision tools were demonstrated to be usable in characterizing behaviors related to mental illness. By using a structured environment, behaviors related to symptoms of a mental illness were elucidated and quantified using computer vision. Novel methods for human computer interaction were developed which facilitate using reconfigurable immersive environments as another avenue for assessment. The impact of these tools can be manifold. Having tools that can collect quantitative data on observed subjects can enable additional objective data. By using immersive environments, enabled by VR and natural interaction, clinicians will be able to use safe, repeatable environments that should elicit discernible responses over the observation period.

While these methods were developed as tools for mental health assessment they can be extended for other uses. Proper HMD and RGB+D sensor calibration has the potential to greatly impact the VR user experience. By being able to reproject the sensed world to the user, they can navigate that space without removing their headset. Having externally affixed RGB+D sensors has its advantages as well. Since the cameras are static traditional background subtraction methodologies or depth thresholding can be employed. This approach also allows the user to interact beyond the view point of the HMD, which is inhibited in the case of HMD mounted RGB+D sensors.

8.1 Contributions

The contributions of this thesis are described here:

- Established an assessment of the state of the art in computer vision, behavior imaging and AR/VR for mental health treatment and assessment.
- Demonstrated the applicability of computer vision methodologies to the mental health domain by being able to classify symptoms related to autism in video.
- Developed a method for assessing OCD related behaviors which used the scenario and environment around the subject to elicit those behaviors.
- Developed a method for registering RGB+D sensors to HMD systems with minimal additional modification.
- Presented a method for extending densely sampled trajectory features for use in scene flow data.
- Provided a framework for further development of mental health assessment scenarios using immersive environments with natural interaction.

8.2 Future Work

This work makes significant strides towards using immersive environments and computer vision to aid in mental health assessment although more work can be done to improve this system. Inexpensive commercially available RGB+D sensors and head-mounted displays have only been around for less than a decade. Their total potential has yet to be seen. There are specific directions relevant to the work of this thesis that warrant further study.

- Pilot Study — This thesis established that environment and scenario can play a role in eliciting behaviors that can then be observed by computer vision. Further validation needs to be done to see if this approach holds when transferred to a virtual environment when interacting with objects that are virtual.

- Scenario Development — A key advantage of this approach is being able to present different scenarios to a subject without changing the essential equipment. Investigating and establishing different scenarios that elicit discernible behaviors will increase the ubiquity of this approach.

- Fine Grain Interaction — Improvements can be made to the point cloud-based interaction methodology. The resolution of the sensor as well as the speed and description of the physics simulation limit the precision in which virtual objects can be manipulated. This has an impact on immersion.

- Activity Detection — Still an important research problem in computer vision, activity detection requires not only determining which activity occurred but when it occurred over the observed sequence.

- Feature Examination — Features extracted along densely sampled trajectories derived from scene flow can be investigated. The advantage of these features is two-fold. Unlike image based methods, trajectories towards the camera can be tracked and distinguishable from noise. Features derived along the trajectory can be computed from 3D data.

- Multiple Sensor Registration — Natural marker less interaction using point cloud data can be improved through incorporating information from multiple views. The precision required for this calibration is significant with respect to other RGB+D sensors as misalignment is easily noticeable in point clouds from two views.

References

[1] World Health Organization. The world health report 2004: Changing history, annex table 3: Burden of disease in DALYs by cause, sex, and mortality stratum in WHO regions, estimates for 2002. *Geneva: WHO*, 2004.

[2] C. Roehrig. Mental disorders top the list of the most costly conditions in the United States: $201 billion. *Health Affairs*, 35(6):1130–1135, June 2016.

[3] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004.

[4] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874. IEEE, June 2014.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, December 2012.

[6] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, December 2015.

[7] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):13.1–13.45, December 2006.

[8] M. Munaro and E. Menegatti. Fast RGB-D people tracking for service robots. *Autonomous Robots*, 37(3):227–242, October 2014.

[9] J. Hashemi, M. Tepper, T. V. Spina, A. Esler, V. Morellas, N. Papanikolopoulos, H. Egger, G. Dawson, and G. Sapiro. Computer vision tools for low-cost and noninvasive measurement of autism-related behaviors in infants. *Autism Research and Treatment*, 2014(1):1–12, June 2014.

[10] J. M. Rehg, G. D. Abowd, A. Rozga, M. Romero, M. A. Clements, S. Sclaroff, I. Essa, O. Y. Ousley, Y. Li, C. Kim, et al. Decoding children's social behavior. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3414–3421. IEEE, December 2013.

[11] S. S. Rajagopalan, A. Dhall, and R. Goecke. Self-stimulatory behaviours in the wild for autism diagnosis. In *IEEE International Conference on Computer Vision Workshops*, pages 755–761. IEEE, December 2013.

[12] A. Ciptadi, M. S. Goodwin, and J. M. Rehg. Movement pattern histogram for action recognition and retrieval. *European Conference on Computer Vision*, pages 695–710, September 2014.

[13] R. A. King, H. Leonard, J. March, et al. Practice parameters for the assessment and treatment of children and adolescents with obsessive-compulsive disorder. *Journal of the American Academy of Child and Adolescent Psychiatry*, 37(10):27S–45S, October 1998.

[14] K. Grabill, L. Merlo, D. Duke, K.-L. Harford, M. L. Keeley, G. R. Geffken, and E. A. Storch. Assessment of obsessive–compulsive disorder: A review. *Journal of Anxiety Disorders*, 22(1):1–17, 2008.

[15] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, January 2013.

[16] K. Kim, C.-H. Kim, S.-Y. Kim, D. Roh, and S. I. Kim. Virtual reality for obsessive-compulsive disorder: Past and the future. *Psychiatry Investigation*, 6(3):115–121, September 2009.

[17] K. Kim, S. I. Kim, K. R. Cha, J. Park, M. Z. Rosenthal, J.-J. Kim, K. Han, I. Y. Kim, and C.-H. Kim. Development of a computer-based behavioral assessment

of checking behavior in obsessive-compulsive disorder. *Comprehensive Psychiatry*, 51(1):86–93, January-February 2010.

[18] L. Scahill, M. A. Riddle, M. McSwiggin-Hardin, S. I. Ort, R. A. King, W. K. Goodman, D. Cicchetti, and J. F. Leckman. Children's Yale-Brown obsessive compulsive scale: Reliability and validity. *Journal of American Academy of Child and Adolescent Psychiatry*, 36(6):844–852, June 1997.

[19] A. S. Radomsky and S. Rachman. Symmetry, ordering and arranging compulsive behaviour. *Behaviour Research and Therapy*, 42(8):893–913, August 2004.

[20] E. Walker, T. Savoie, and D. Davis. Neuromotor precursors of schizophrenia. *Schizophrenia Bulletin*, 20(3):441–451, January 1994.

[21] J. Schiffman, E. Walker, M. Ekstrom, F. Schulsinger, H. Sorensen, and S. Mednick. Childhood videotaped social and neuromotor precursors of schizophrenia: a prospective investigation. *American Journal of Psychiatry*, 161(11):2021–2027, November 2004.

[22] E. M. Mahone, D. Bridges, C. Prahme, and H. S. Singer. Repetitive arm and hand movements (complex motor stereotypies) in children. *Journal of Pediatrics*, 145(3):391–395, September 2004.

[23] S. Goldman, C. Wang, M. W. Salgado, P. E. Greene, M. Kim, and I. Rapin. Motor stereotypies in children with autism and other developmental disorders. *Developmental Medicine and Child Neurology*, 51(1):30–38, January 2009.

[24] H. Singer. Motor stereotypies. *Seminars in Pediatric Neurology*, 16(2):77–81, June 2009.

[25] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, September 2005.

[26] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. *European Conference on Computer Vision*, pages 650–663, October 2008.

[27] A. Castrodad and G. Sapiro. Sparse modeling of human actions from motion imagery. *International Journal of Computer Vision*, 100(1):1–15, October 2012.

[28] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3169–3176. IEEE, June 2011.

[29] H. Wang, M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference*, pages 124.1–124.11. BMVA Press, September 2009.

[30] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893. IEEE, June 2005.

[31] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *European Conference on Computer Vision*, pages 428–441, May 2006.

[32] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, June 2008.

[33] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, June 2010.

[34] X. Wang, L. Wang, and Y. Qiao. A comparative study of encoding, pooling and normalization methods for action recognition. In *Asian Conference on Computer Vision*, volume 7726. Springer, November 2012.

[35] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, October 2013.

[36] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *International Workshop on Human Behavior Understanding*, pages 29–39. Springer, November 2011.

[37] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, March 2013.

[38] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *IEEE International Conference on Computer Vision*, pages 4489–4497. IEEE, December 2015.

[39] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732. IEEE, June 2014.

[40] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, December 2014.

[41] G. Gkioxari and J. Malik. Finding action tubes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 759–768. IEEE, June 2015.

[42] I. Laptev, S. Belongie, P. Perez, and J. Wills. Periodic motion detection and segmentation via approximate sequence alignment. In *IEEE Conference on Computer Vision*, pages 816–823. IEEE, October 2005.

[43] P. Wang, G. D. Abowd, and J. M. Rehg. Quasi-periodic event analysis for social game retrieval. In *IEEE Conference on Computer Vision*, pages 112–119. IEEE, October 2009.

[44] R. Cutler and L. Davis. View-based detection and analysis of periodic motion. In *IEEE Conference on Pattern Recognition*, pages 495–500. IEEE, August 1998.

[45] I. Junejo, E. Dexter, I. Laptev, and P. Pérez. View-independent action recognition from temporal self-similarities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):172–185, March 2011.

[46] H. Kato and M. Billinghurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *IEEE and ACM International Workshop on Augmented Reality*, pages 85–94. IEEE, October 1999.

[47] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, November 2001.

[48] E. Suma, D. M. Krum, and M. Bolas. Sharing space in mixed and virtual reality environments using a low-cost depth sensor. In *IEEE International Symposium on VR Innovation*, pages 349–350. IEEE, March 2011.

[49] Ovrvision — USB3.0 stereo camera for Oculus Rift, OSVR. `http://ovrvision.com/` [Online; accessed 8-December-2015].

[50] S. M. LaValle, A. Yershova, M. Katsev, and M. Antonov. Head tracking for the oculus rift. In *International Conference on Robotics and Automation*, pages 187–194. IEEE, May 2014.

[51] O. Hilliges, D. Kim, S. Izadi, M. Weiss, and A. Wilson. Holodesk: direct 3D interactions with a situated see-through display. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 2421–2430. ACM, May 2012.

[52] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *European Conference on Computer Vision*, pages 25–36, May 2004.

[53] B. Jones, R. Sodhi, M. Murdock, R. Mehra, H. Benko, A. Wilson, E. Ofek, B. MacIntyre, N. Raghuvanshi, and L. Shapira. Roomalive: Magical experiences enabled by scalable, adaptive projector-camera units. In *ACM Symposium on User interface Software and Technology*, pages 637–644. ACM, October 2014.

[54] B. O. Rothbaum, A. A. Rizzo, and J. Difede. Virtual reality exposure therapy for combat-related posttraumatic stress disorder. *Annals of the New York Academy of Sciences*, 1208(1):126–132, October 2010.

[55] N. Josman, E. Somer, A. Reisberg, P. L. Weiss, A. Garcia-Palacios, and H. Hoffman. BusWorld: Designing a virtual environment for post-traumatic stress disorder in israel: A protocol. *Cyberpsychology and Behavior*, 9(2):241–244, April 2006.

[56] J. Difede and H. G. Hoffman. Virtual reality exposure therapy for world trade center post-traumatic stress disorder: A case report. *Cyberpsychology and Behavior*, 5(6):529–535, July 2002.

[57] T. D. Parsons and A. A. Rizzo. Affective outcomes of virtual reality exposure therapy for anxiety and specific phobias: A meta-analysis. *Journal of Behavior Therapy and Experimental Psychiatry*, 39(3):250–261, September 2008.

[58] D. Schroeder, F. Korsakov, J. Jolton, F. J. Keefe, A. Haley, and D. F. Keefe. Creating widely accessible spatial interfaces: Mobile VR for managing persistent pain. *IEEE Computer Graphics and Applications*, 33(3):82–88, May-June 2013.

[59] O. A. Van den Heuvel, D. J. Veltman, H. J. Groenewegen, R. J. Dolan, D. C. Cath, R. Boellaard, C. T. Mesina, A. J. Van Balkom, P. Van Oppen, M. P. Witter, A. A. Lammertsma, and R. van Dyck. Amygdala activity in obsessive-compulsive disorder with contamination fear: A study with oxygen-15 water positron emission tomography. *Psychiatry Research: Neuroimaging*, 132(3):225–237, December 2004.

[60] D. Simon, E. Kischkel, R. Spielberg, and N. Kathmann. A pilot study on the validity of using pictures and videos for individualized symptom provocation in obsessive–compulsive disorder. *Psychiatry Research*, 198(1):81–88, June 2012.

[61] P. Nolin, A. Stipanicic, M. Henry, Y. Lachapelle, D. Lussier-Desrochers, A. A. Rizzo, and P. Allain. ClinicaVR: Classroom-CPT: A virtual reality tool for assessing attention and inhibition in children and adolescents. *Computers in Human Behavior*, 59:327–333, June 2016.

[62] J. Cegalis. *VIGIL: Software for testing concentration and attention.* Forthought Ltd., Nashua, NH, 1991.

[63] T. D. Parsons, T. Bowerly, J. G. Buckwalter, and A. A. Rizzo. A controlled clinical comparison of attention performance in children with adhd in a virtual reality classroom compared to standard neuropsychological methods. *Child Neuropsychology*, 13(4):363–381, 2007.

[64] U. Díaz-Orueta, C. Garcia-López, N. Crespo-Eguílaz, R. Sánchez-Carpintero, G. Climent, and J. Narbona. AULA virtual reality test as an attention measure: Convergent validity with conners continuous performance test. *Child Neuropsychology*, 20(3):328–342, 2014.

[65] T. D. Parsons and A. R. Carlew. Bimodal virtual reality stroop for assessing distractor inhibition in autism spectrum disorders. *Journal of Autism and Developmental Disorders*, 46(4):1255–1267, 2016.

[66] T. D. Parsons, S. McPherson, and V. Interrante. Enhancing neurocognitive assessment using immersive virtual reality. In *IEEE Workshop on Virtual and Augmented Assistive Technology*, pages 27–34. IEEE, March 2013.

[67] J. Fasching, N. Walczak, R. Sivalingam, K. Cullen, B. Murphy, G. Sapiro, V. Morellas, and N. Papanikolopoulos. Detecting risk-markers in children in a preschool classroom. In *IEEE Conference on Intelligent Robots and Systems*, pages 1010–1016. IEEE, October 2012.

[68] A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnörr. Real-time optic flow computation with variational methods. In *International Conference on Computer Analysis of Images and Patterns*, pages 222–229. Springer, August 2003.

[69] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, November 2006.

[70] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(1):19–60, March 2010.

[71] T. Guha and R. Ward. Learning sparse representations for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1576 –1588, August 2012.

[72] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, April 2004.

[73] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, April 2011.

[74] M. Mostafa. An architecture for autism: Concepts of design intervention for the autistic user. *International Journal of Architectural Research*, 2:189–211, March 2008.

[75] D. L. Schilling and I. S. Schwartz. Alternative seating for young children with autism spectrum disorder: Effects on classroom behavior. *Journal of Autism and Developmental Disorders*, 34(4):423–432, August 2004.

[76] G. A. Bernstein, T. Hadjiyanni, K. R. Cullen, J. W. Robinson, E. C. Harris, A. D. Young, J. Fasching, N. Walczak, S. Lee, V. Morellas, and N. Papanikolopoulos. Use of computer vision tools to identify behavioral markers of pediatric obsessive–compulsive disorder: A pilot study. *Journal of Child and Adolescent Psychopharmacology*, 27(2):140–147, March 2017.

[77] J. Yao and J. Odobez. Multi-layer background subtraction based on color and texture. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, June 2007.

[78] T. Bouwmans, F. El Baf, and B. Vachon. *Handbook of Pattern Recognition and Computer Vision*, chapter 2.3: Statistical background modeling for foreground detection: A survey, pages 181–189. World Scientific, 4th edition, 2010.

[79] S. Brutzer, B. Höferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1937–1944. IEEE, June 2011.

[80] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, July 2012.

[81] T. O'Haver. A pragmatic introduction to signal processing, 1997.

[82] PCL, 2015. `http://pointclouds.org/` [Online; accessed 12-September-2015].

[83] OpenCV, 2015. `http://opencv.org/` [Online; accessed 12-September-2015].

[84] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[85] J. J. Craig. *Introduction to robotics: mechanics and control*. Pearson Prentice Hall, 3rd edition, 2005.

[86] J.-Y. Bouguet. Camera calibration toolbox for matlab. *Computational Vision at the California Institute of Technology*, 2004.

[87] C. Gramkow. On averaging rotations. *Journal of Mathematical Imaging and Vision*, 15(1):7–16, July 2001.

[88] D. Q. Huynh. Metrics for 3D rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, October 2009.

[89] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, June 2014.

[90] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[91] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *IEEE International Conference on Computer Vision*, volume 2, pages 722–729. IEEE, September 1999.

[92] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *IEEE International Conference on Computer Vision*, pages 1–7. IEEE, October 2007.

[93] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers. Stereoscopic scene flow computation for 3D motion understanding. *International Journal of Computer Vision*, 95(1):29–51, October 2011.

[94] M. Jaimez, M. Souiai, J. Gonzalez-Jimenez, and D. Cremers. A primal-dual framework for real-time dense rgb-d scene flow. In *IEEE International Conference on Robotics and Automation*, pages 98–104. IEEE, May 2015.

[95] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An algorithm for minimizing the Mumford-Shah functional. In *IEEE International Conference on Computer Vision*, pages 1133–1140. IEEE, October 2009.

[96] M. Macklin, M. Müller, N. Chentanez, and T.-Y. Kim. Unified particle physics for real-time applications. *ACM Transactions on Graphics*, 33(4):153, July 2014.

[97] NVIDIA FLeX — developer, 2017. `https://developer.nvidia.com/flex` [Online; accessed 07-March-2017].

[98] S. Green. CUDA particles. *NVIDIA whitepaper*, 2(3.2):1, 2008.

[99] C. Everitt. Interactive order-independent transparency. *NVIDIA whitepaper*, 2(6):7, 2001.